MONTAN
UNIVERSITÄT
LEOBEN

Chair of Reservoir Engineering

Master's Thesis

Primary drainage in static pore network modelling: a comparative study

Omidreza Amrollahinasab Mahdiabad

September 2020

Dedicated to my family and friends
who were always there to support me.

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been published elsewhere. This dissertation is the outcome of my own work using only cited literature.

Erklärung

Hiermit erkläre ich, dass der Inhalt dieser Dissertation, sofern nicht ausdrücklich auf die Arbeit Dritter Bezug genommen wird, ursprünglich ist und nicht an anderer Stelle veröffentlicht wurde. Diese Dissertation ist das Ergebnis meiner eigenen Arbeit mit nur zitierter Literatur.

Omidreza Amrollahinasab Mahdiabad, 02 September 2020

Acknowledgements

I would like to express my sincere gratitude to my supervisor Prof. Holger Ott and my co-supervisor, Pit Arnold, for the continuous support of my master's study and research. Their guidance helped me in all time during my research and writing of this thesis.

Besides, I would like to thank my industrial supervision Jan Steckhan and OMV for their excellent support during my master thesis research and providing me with the necessary facilities.

I also place on record my sense of gratitude to one and all who, directly or indirectly, have lent their helping hand in this venture.

Abstract

Darcy's law or other similar approaches are implemented to model the flow at the continuum scale. However, the accuracy of these models depends on porous media parameters such as porosity, relative permeability, capillary pressure relationships, etc (Bhattad et al., 2011). One option is to obtain these parameters experimentally, but in recent years, the development of rock imaging tools that image the porous rock and the fluids inside them has improved our understanding of the flow in porous media. In combination with publicly available numerical tools, it enables us to simulate the flow in these media (J. Blunt, 2017).

Pore network modeling (PNM) is a widely used technique to simulate multiphase transport in porous materials, and one of the open-source available software packages to do so is OpenPNM. Several porous media research groups have developed it as a pore network modeling tool. In pore network modeling, a pore network is extracted from reconstructed microtomography images and the corresponding numerical calculations are done to predict the transport properties of the porous media (Gostick J. A., 2016).

Pore network modeling leads to the calculation of the petrophysical and advanced rock flow properties. In this work, the OpenPNM python package is implemented and used to simulate the primary drainage capillary pressure and relative permeability from microtomography images of sandstone rocks. Pore networks are extracted using the Porespy python package which uses an algorithm named SNOW to extract pore network from segmented microtomography rock images.

The model is validated using the experimental results of capillary pressure, absolute permeability, and primary drainage relative permeabilities from well-known rock samples. Simulation results have proved to match the experimentally obtained results within an acceptable range of uncertainty. A sensitivity analysis is then conducted to understand the influence of modeling parameters, interfacial tension, and contact angle on the results. Finally, a workflow is presented to predict the petrophysical and transport properties of unknown rock samples. The implementation of the workflow is then demonstrated on a sandstone rock of interest.

Zusammenfassung

Darcy's law und andere vergleichbare Methoden wurden zur Modellierung der Strömung auf Kontinuumsskala entwickelt. Die Genauigkeit dieser Modelle hängt von Parametern wie der Porosität, der relativen Permeabilität oder dem Kapillardruck ab (Bhattad et al., 2011). Eine Möglichkeit ist es diese Paramter experimentell zu bestimmen, in den letzten Jahren jedoch, hat die Entwicklung von Bildgebenden Verfahren, welche das poröse Gestein und die sich darin befindenden Flüssigkeiten abbilden, unser Verständniss von Strömungen in porösen Medien weiterentwichelt. In Verbindung mit öffentlich zugänglichen numerischen Werkzeugen ermöglicht dies die Simulation von Strömungsvorgängen in diesen Materialien (J. Blunt, 2017).

Poren Netzwerk Modellierung (PNM) ist eine häufig eingesetzte Methode zur Simulation von mehrphasigem Transport in porösen Materialien. Ein open-source Software für derartige Simulationen ist OpenPNM. Es wurde von mehreren Forschungsgruppen zur Poren Netzwerk Modellierung entwickelt. Ein Poren Netzwerk wird dabei aus rekonstruierten Mikrotomographie Bildern extrahiert und die entsprechenden numerischen Berechnungen ermöglichen die Vorhersage von Transporteigenschaften der porösen Materialien (Gostick J. A., 2016).

Poren Netzwerk Modellierung erlaubt die Berechnung von petrophysikalischen und komplexen Strömungseigenschaften im Gestein. Die vorliegende Arbeit nutzt das OpenPNM Python Paket zur Simulation des primären Kapillardrainage Druckes und der relativen Permeabilität von Mikrotomographie Daten eines Sandgesteines. Das Python Paket Porespy nutzt dabei den SNOW Algorithmus zur Extrahierung des Poren Netzwerkes aus segmentierten Micro-tomographie Bildern.

Das Modell wurde mit experimentellen Daten für Kapillardruck, Absolute Permeabilität und primäre Drainage relative Permeabilität von sehr bekannten Gesteinsproben validiert. Simulationsergebnisse haben, unter Einbezug eines aktzeptablen Ungenauigkeitsbereichs, eine gute Übereinstimmung mit den experimentellen Ergebnissen aufgewiesen. Des Weiteren wurde eine Sensitivitätsanalyse durchgeführt um ein besseres Verständnis des Einflusses der Modelparameter, wie Grenzflächenspannung und Kontaktwinkel, auf die Ergebnisse zu erhalten. Abschließend wird ein Arbeitsablauf zur Vorhersage von petrophysikalischen- und Transporteigenschaften von unbekannten Gesteinen präsentiert. Die Implementierung des Arbeitsablaufes wurde exemplarisch an einem Sandgestein durchgeführt.

Table of Contents

Chapter 1

Introduction

Some of the main problems of our century are securing access to clean water for drinking and agriculture, providing enough energy for the growing human population, and tackling the issue of climate change at the same time. To be able to solve these problems, we need a comprehensive understanding of flow in porous media to quantify how fluids move through porous rock. This requirement is because most of the available freshwater lies within underground resources, and most of our energy comes from oil and gas resources, which also come from underground reservoirs (J. Blunt, 2017).

To develop and produce from oil and gas reservoirs, we use the guidance of our reservoir models, and the accuracy of our reservoir models depend on the relative permeabilities and proper fluid modeling. One option to derive these relative permeabilities would be to use special core analysis in a laboratory. The alternative is to use the Digital Rock Physics (DRP) approach in which numerical core analysis is done on a computer. The advantage is that it saves us time and costs, and the same samples can be investigated under different scenarios (Geodict, 2020).

Darcy's law or other similar approaches can be implemented to model the flow at the continuum scale. However, the accuracy of their models depends on porous media parameters such as porosity, relative permeability, capillary pressure relationships, etc. (Bhattad et al., 2011). These parameters are obtained experimentally, but in recent years, the development of rock imaging tools that image the porous rock and the fluids inside them has improved our understanding of the flow in porous media. This tool is combined with publicly available numerical tools which allow us to simulate the flow in these media (J. Blunt, 2017).

There are two approaches available to model the flow in the pore scale. The first approach, which is called direct numerical simulation, solves the governing equations on the actual geometry of the pore space. The problem with this approach is that it is very computationally expensive. The other method is to use pore network modeling, in which the pore space is represented with a network of balls and sticks. This kind of modeling is highly efficient at the expense of losing information and the accuracy of predictions (Geodict, 2020).

Pore network modeling is a widely used technique to simulate multiphase transport in porous materials, and one of the available software packages is OpenPNM. Several porous media research groups have developed it as a pore network modeling tool. In pore network modeling, a pore network is extracted from reconstructed microtomography images and the corresponding numerical calculations are done to predict the transport properties of the porous media (Gostick J. A., 2016).

The input for the model is a high-resolution segmented image of the rock's pore structure obtained by micro-computed tomography imaging tools. This leads to the digital twin from which the petrophysical and advanced flow properties can be calculated. Elementary fluid displacement processes and fluid phase distribution are obtained using micro-CT imaging and microfluidic flow experiments in combination with numerical flow simulations.

Direct numerical simulations are mostly used in the case that porous media is considered as a volume averages continuum without the determination of the micro-scale features. Continuum models have some practical limitations because they are mathematically hard. For example, to describe the porous media's macroscopic transport properties, experiments have to be conducted to measure the appropriate relationships. These properties can be e.g., permeability coefficient or effective diffusivity, and they are not easy to measure, especially if there is multiphase flow. The second practical limitation is that, in a volume-averaged continuum modeling, the discrete pore-scale events are not determined; therefore, the model is calculating the average amount of fluids. As a result, the models that depend on extensions of Darcy's law for multiphase flow cannot predict the distribution of the fluids inside the continuum, and we need more comprehensive formulations for this. Pore network modeling tackles this issue; however, it takes away the challenging mathematical formulations (Gostick J. A., 2016).

Pore network modeling (PNM) is an alternative for direct numerical simulation (DNS), which offers several advantages by making simplifying assumptions. PNM is much less computationally expensive than DNS, and it can manage millions of pores, whereas DNS can only handle thousands. Moreover, similar to DNS, static multiphase flow simulations can be easily done with PNM using, e.g., percolation theory (J. Gostick et al., 2017). In pore network modeling, the pore space is separated into pores and the throats which are connecting the pores. Then the flow in the pore throats is calculated by applying mass balance equations on each pore and solving Poiseuille-type equations. These types of simulations have been used to study drainage, imbibition, single, and relative permeability processes in porous media (Bhattad et al., 2011).

In this work, the OpenPNM python package, which is an open-source pore network modeling package presented by (Gostick J. A., 2016) is implemented to simulate the capillary pressure and relative permeabilities on models of microtomography images. The pore network extraction is done using the Porespy python package (J. Gostick et al., 2017), which is using an algorithm named SNOW to extract pore network from microtomography rock images. The reason for this choice is the features and flexibility of the algorithm in the framework work of OpenPNM package.

As a result, it is needed to validate and understand the algorithms, which OpenPNM is using, to simulate and predict physical rock properties. Therefore, sensitivity analysis is run to investigate the effect of pore network extraction parameters and rock and fluid properties ,e.g. interfacial tension and contact angle, on the prediction results.

In the end, based on the observations made, a workflow is proposed to predict the flow parameters of porous media using pore network modeling. Implementing this workflow, capillary pressure, absolute permeability, and relative permeability of the porous media can be predicted. The implementation of this workflow is then demonstrated on a sandstone rock of interest.

Chapter 2

State of the Art

2.1 Pore Space Imaging

Our ability to investigate transport in porous media has been transformed by the use of pore space imaging methods. Such methods produce a 2D image of the pore space using X-rays and then reconstruction software are used to construct 3D models of the pore space. The principle behind these imaging tools are the same as the computer tomography (CT) scanning devices which are used for medical examinations, and as a result, medical scanners are adapted to scan the cores that have a diameter in centimeters and are 1-2 meters long. However, if the images are taken from cores with such dimensions, then the image resolution is in the lower mm range, which is not sufficient enough to see the actual pore structure of the rock. Therefore, smaller samples have to be imaged at a closer distance to the X-ray source, which will produce images at the scale of micrometers (J. Blunt, 2017).

Since the first micron-resolution image produced by Flannery in 1987, the technology has developed tremendously, and nowadays, universities and laboratories have their own imaging scanners and access to central synchrotron facilities. Different X-ray tomography schematics are illustrated in Figure 1 (J. Blunt, 2017).

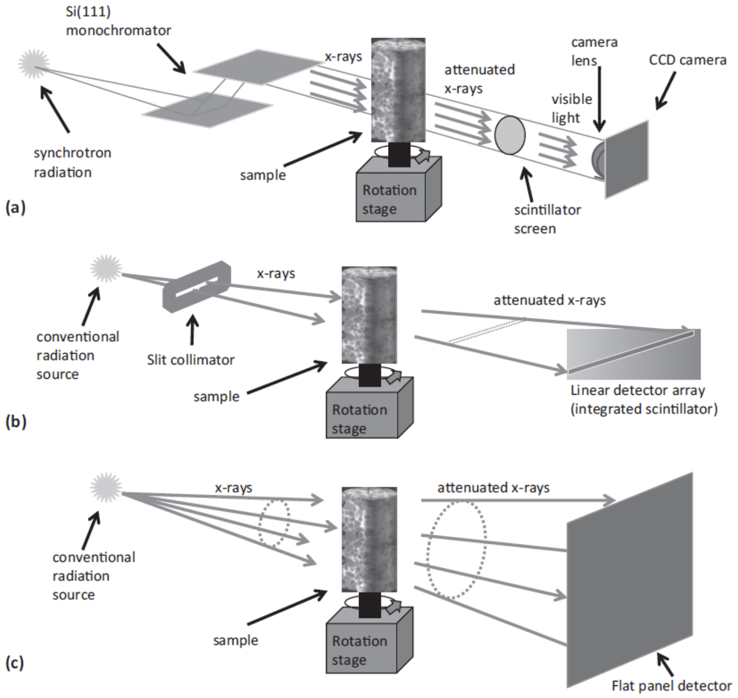


Figure 1: Different X-ray tomography schematics (a) source, sample, and receiver at a synchrotron beam-line (b) micro-CT system with a fan-beam (c) cone-beam configuration (J. Blunt, 2017)

After the X-rays are emitted from the X-ray source, the radiation is absorbed by the rock more strongly than the fluids or the air-filled pore space. Therefore, implementing the correct experimental design, the rock and pore structures are distinguishable. The resolution of the resulting image depends on sample size, beam quality, and detector specifications. In the case of a cone-beam setup, the resolution also depends on the sample distance to the beam source. Smaller samples can be brought closer to the source and therefore allow lower resolutions. As a result, based on the detector, the sample sizes at the scale of millimeters are typically scanned at micrometer resolutions, resulting in an image resolution of micrometers with 1000^3 to 2000^3 voxels. It is also possible to image the rocks under high temperatures and pressure, which can simulate deep underground conditions (J. Blunt, 2017). A micro-CT image of Bentheimer sandstone is demonstrated in Figure 2 (Ramstad et al., 2012).

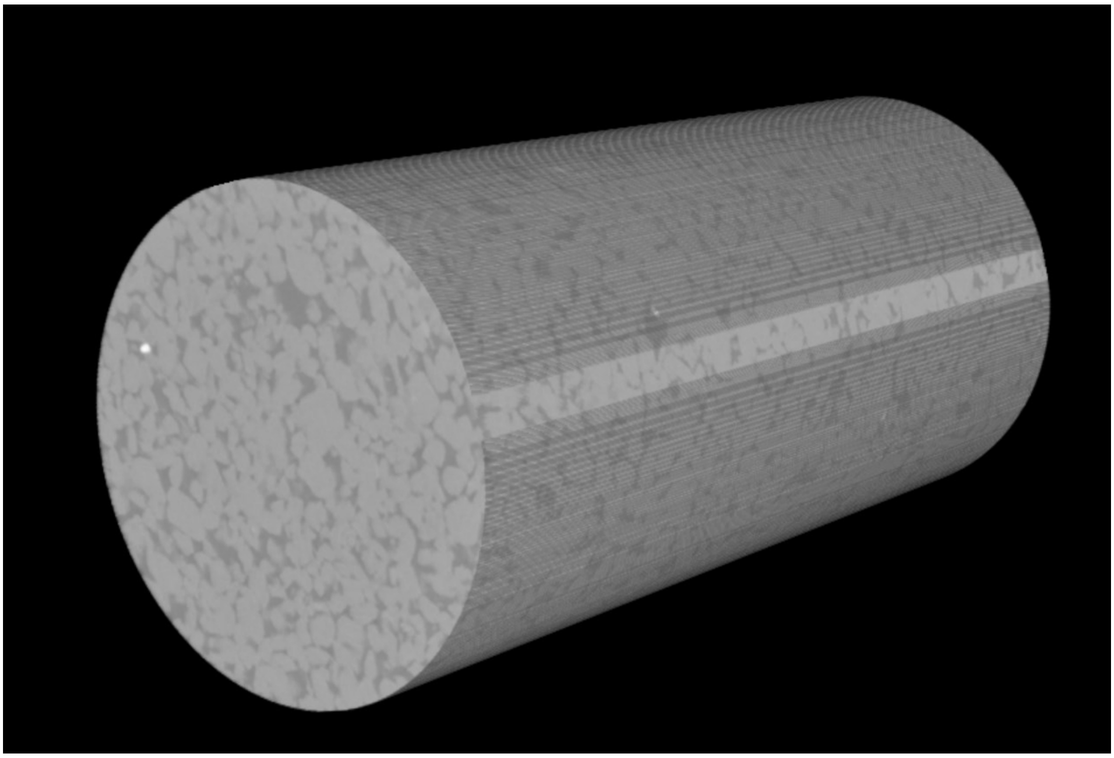


Figure 2: 3D reconstruction of micro-CT image of Bentheimer sandstone (Ramstad et al., 2012)

2.1.1 Statistical and Process-Based Pore Networks

Statistical and process-based methods have been implemented to create representations of the pore space and to derive the essential features of the porous medium, e.g., porosity and connectivity. The benefit of these models is that they do not have limitations in terms of resolution of size. Moreover, different realizations can be created to analyze the variation of properties. There are two types of process-based models (J. Blunt, 2017).

In the first type of process-based modeling, the rock's pore network is constructed by mimicking the sedimentary and diagenetic processes. It was first developed by Bakke & Øren (1997). They tried to reconstruct a sandstone and compute the flow properties by analyzing the pore size distribution of two-dimensional thin sections. The flow properties which were derived from this approach were close to the ones derived from direct simulation on the images of the rock (Øren et at., 2007).

In another approach, geo-statistics is implemented to create a discretized image of the pore space. In this method, single and two-point correlation functions of the pore space are made using a high-quality two-dimensional training image. The problem with this method is that the long-range connectivity of pore space is not well represented (J. Blunt, 2017; Adler, 1990).

Other statistical measures have been implemented to preserve the connectivity of the pore space. Generally, to maintain the essential geometric and flow properties, any information can be introduced into a statistical reconstruction. Moreover, three-dimensional images from micro-CT scans can be used as a benchmark to condition the statistical reconstruction (J. Blunt, 2017). For example, Biswal et al. generated a synthetic representation of Fontainebleau sandstone using a model of grain packing Biswal et al. (1999). In this representation, the angular nature of some of the grains was lost, but the connectivity of the pore space was preserved. Furthermore, more complex shapes were produced, but then the connectivity was not preserved.

Using this statistical reconstruction, images are produced with resolutions beyond the capabilities of the imaging techniques. This resolution is required to capture the geometry of pore space of different rocks from simple sandstone rocks to micro porosity in carbonates and shales. Although these reconstructions overcome some physical limitations, they require further simplification to quantify flow and transport in these models (J. Blunt, 2017).

2.2 Pore network extraction and construction

A pore network is extracted either from a micro-CT image or from a statistical reconstruction of the pore space. Such a network includes wider regions as pores, and narrower regions called throats, which connect the pores to each other. The goal of network construction is to characterize the pore space in a way that allows us to capture the critical features of flow and fluid displacement precisely (J. Blunt, 2017).

The first representation of a pore space was constructed by Finney (1970) who equally sized sphere-shaped ball bearings in a balloon to create a random packing. In this construction, ball bearings represented the grains, and the regions with the highest distance from this grain centers were identified as pores, while the voids connecting these pores were identified as throats. After that, other researchers tried to develop this approach by accounting for compaction and diagenesis effects (Bryant & Blunt, 1992; Bryant et al., 1993a; Bryant, 1993b). To mimic the compaction, the grains were pushed closer together in a direction, and they swelled to mimic the diagenesis effects. Furthermore, Bakke and Øren extended the approach by developing numerical methods to perceive sedimentary and diagenetic effects (Bakke & Øren, 1997; Øren et al., 1998). In their model, after the construction of a network in the same way as Bryant in 1993, numerical models were developed for settling, compaction, and sedimentation of the grains (Bryant, 1993b). Several networks were constructed using this approach, from which successful predictions were made. A synthetic pore space image of the Berea sandstone, with its corresponding extracted pore network, is demonstrated in Figure 3 (J. Blunt, 2017).

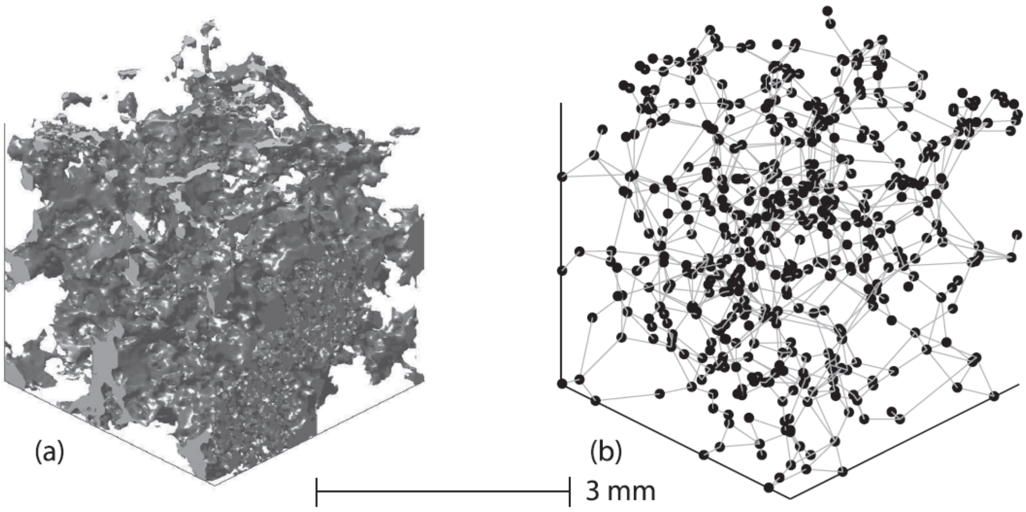


Figure 3: (a) Synthetic reconstruction of Berea sandstone using the process-based method (b) corresponding extracted pore network (Bakke & Øren, 1997)

In the process-based method, pores and throats will have angular cross-sections which come from the structure of the pore space. This angularity is taken into consideration by implementing the shape factor from Mason & Marrow (1991). They defined the shape factor as G, which is the cross-sectional area A, divided by square perimeter P.

$$G = \frac{A}{P^2} \tag{2.1}$$

Using this shape factor, the inscribed radius can be preserved for the calculation of threshold capillary pressure and the wetting phase, which lies at the corners as a wetting film. This method is limited to the cases when the formation of the rock itself can be simulated or when quantification of the grain geometries is available. This can be challenging for the rocks with the biogenic origin, which have undergone diagenesis processes like carbonates.

Therefore, a more topological approach is required for general network extraction from the pore space. One of the early models is the Medial axis skeletonization model. In this model, the medial axis of the pore space, which had the greatest distance from the solid surface, was used to represent the pore space. This was done by an iterative process in which the pore space was shrunk by eliminating the voxels that were next to a solid and in the end the three-dimensional pore space was converted to a graph of medial voxels (Thovert et al., 1993; Spanne et al., 1994; Lindquist et al., 1996). While this model preserves the topology of the pore space, it has some disadvantages. For example, in real cases, the identification of the pores is dependent on the order that the pore space is shrunk, which can be a big problem since there is a trade-off between the image resolution and sample size. Moreover, some features that were not relevant for the flow were captured like dead ends, or the identification of the pores was less straight forward. Despite the limitations, the model was tested on several rock samples and demonstrated accurate predictions of flow and transport properties (Prodanovi´c et al., 2007; Yang et al., 2015).

Due to the problems of the topological method, a robust alternative was proposed to identify the pores and put an emphasis on finding wide and narrow regions of the pore space. In this alternative, the pores were grown in pore space voxels until they hit the solid, which are called maximum balls. In a series of overlapping spheres, a pore is defined as the largest sphere, which is called an ancestor. Spheres that are completely contained in other spheres are ignored. Smaller maximum balls that have overlapped with an ancestor are assigned to the ancestor's family. These assignments to the ancestor's family will lead to a ball, which is called a child of two families, and then the throats are defined. Here also, the lines which connect the centers of maximum balls are the medial axis skeleton (Silin & Patzek, 2006). Figure 4 illustrates how the pores and throats are formed in the maximum ball method.

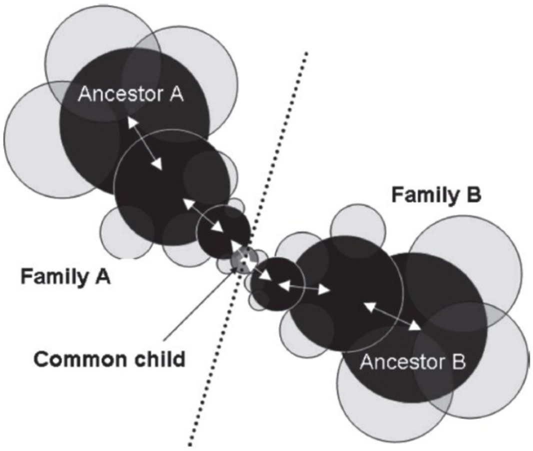


Figure 4: Maximum ball method (Dong & Blunt, 2009)

However, there are also problems existing in this method as well. Extracting a network using this method can cause the formation of small pore and throats which do not contribute to the flow, and it is limited by the ambiguities of the discrete nature of pore images. Also, problems exist in this method regarding the distinction between the pores and throats. Nevertheless, attempts were made to combine this method with simplifications of the pore space, using shape factors in equation (2.1), resulting in successful predictions of the flow behavior (Dong & Blunt, 2009). Since the maximum ball method has a better representation of the pores and the medial axis method has a better description of the throats, it is also possible to mix the two approaches to extract the pore network. This technique was implemented on unconsolidated media with the results matching the experimental data (Al-Raoush, 2005).

Another method to extract the pore network is based on image analysis. In this technique, at first, a distance map of the pore space is constructed, which is the distance of the pore voxels to the nearest solid surface. The maximums in this distance map are inserted into a seeded watershed algorithm to segment the pore space into pores and throats. These maximums represent the pores, and the distance to the nearest solid surface defines the pore radius, which is equivalent to the ancestor maximum ball in the previous method. The watershed algorithm is standard in hydrology to divide regions into different drainage basins and define where the water will eventually flow. The analogy here is that the distance in the distance map represents the depth. The pores will be located downhill where the distance is at its maximum, and the throats will be situated at minimums in the distance map which represent higher grounds (Wildenschild & Sheppard, 2013; Rabbani et al., 2014; Taylor et al., 2015; Prodanovi´c et al., 2015).

After segmentation of the image into different regions, all the volume of the pore space is assigned to pores and throats are defined at the boundaries between the regions. It is important to note that these boundaries between the regions are not necessarily planar surfaces. The problems with this technique are that in sheet-like pore space, the distance map is not a good representation of the segmented pore space. Moreover, the same two pores might be connected with more than one throat in between (J. Blunt, 2017).

Bhattad et al. (2011) has done a comprehensive study to investigate the effect of different pore network structures on the modeling results. They conclude that even if the networks are having orders of magnitudes of difference in pore size distribution and pore connectivity, single-phase permeability predictions remain consistent due to the implementation of physically representative network models. Capillary pressure is, on the other hand, more sensitive to the network's underlying structure. However, the pore size distribution obtained from these capillary pressures is again consistent in different networks.

One of the difficulties in extracting a pore network model from a pore space image is the unclarity in the physical and geometrical definition of what is pore and what is pore throat. This uncertainty prevents the existence of a universal pore structure model for different materials. Therefore several network structures could be extracted from pore space data, and there is no straightforward way of identifying the correct one (Bhattad et al., 2011).

Baychev (2019) also studied the performance of the most used algorithms, namely maximum ball (Dong & Blunt, 2009) and watershed segmentation (Rabbani et al., 2014). Their findings conclude that maximum ball can identify a larger number of features compared to the watershed segmentation algorithm, and over-segmentation is an important problem for the watershed segmentation algorithm. However, the maximum ball method may introduce conductive paths that do not exist in the original image. In general, the maximum ball method is generating smaller features compared to the watershed segmentation algorithm. In both algorithms, the number of the pores and throats are overestimated, and the average coordination number is underestimated. Despite the underlying structural difference, permeability is predicted with acceptable accuracy by both algorithms, which is also in agreement with the findings of (Bhattad et al., 2011) (Baychev, 2019).

(Rabbani et al., 2014) extraction method mainly had the problem that in materials with higher porosity, it resulted in many spurious peaks. This was due to the peaks in the distance transform map at plateaus and ridges along with the pore centers, which resulted in an over segmented image. J. Gostick et al. (2017) approached to improve this method by adding filters that find the spurious peaks and eliminate them. In this algorithm, at first, the over segmented image is created using the watershed algorithm, and then the number of peaks is ruled out progressively, resulting in a subnetwork of the over segmented watershed (SNOW).

2.3 Direct Numerical Simulation

In the direct numerical simulation approach, the governing flow equations are solved on the actual pore geometry, which is obtained through microtomography imaging (Geodict, 2020). The PNM is rule based whereas DNS solves differential equations which are based on physics. One of the available tools to perform direct numerical simulations in porous media is Geodict from Math2Market. Geodict can import microtomography scans and analyze the porous media using its PoroDict module. PoroDict can characterize the pore space by determining the geometric pore size distribution, pore size distribution by porosimetry, and percolation path.

Geometric pore size distribution is a purely geometrical method in which pore radius is determined by fitting spheres into the pore volume. In this method, through pores, closed pores, and blind pores are not distinguished. The equivalent to experimental porosimetry methods, such as MICP is pore size distribution by porosimetry. In porosimetry, the volume of the non-wetting fluid, which has invaded the porous media, is calculated. This method is similar to geometric pore size distribution, however, in porosimetry, the invading phase must be connected to the inlet, and the closed pores are also accounted for. The percolation path is calculated as the shortest path between the inlet and the outlet, in which a spherical particle with the maximum possible diameter can travel (Geodict, 2020).

In porosimetery the connectivity of the pore space is considered because laboratory porosimetry tests e.g. MICP, the isolated pores are not measured. Figure 5 demonstrates how PoroDict calculates porosimetry. On the left of this figure, the solid material is shown with white dots, and the pore space is black. On the right of this figure, pores with diameters more than 40 μm are shown in dark grey. The pores with diameters between 32-40 μm are shown in light grey (Planas & Rief, 2019).

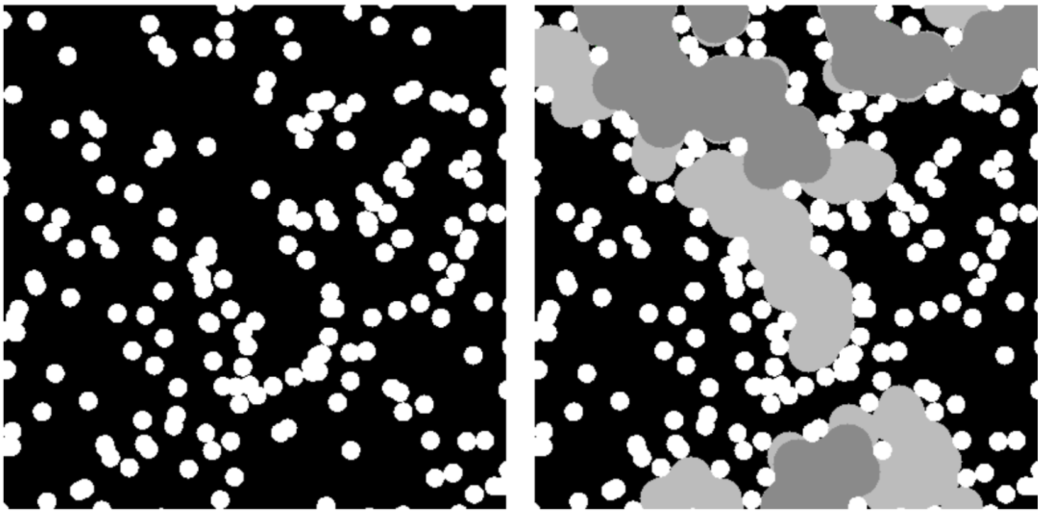


Figure 5: Computation of porosimetry in PoroDict (Planas & Rief, 2019)

After the characterization of the pore space, the flow behavior of the pore space is predicted with the FlowDict module. FlowDict can predict the flow velocity and permeability of the pore space by simulating the flow experiments and process the simulation results. The module uses Darcy's law to predict the permeability of the pore space using flow velocity, pressure drop, fluid viscosity, and the length of the porous media. These calculations apply to the Stokes flow in which the flow velocity and Reynolds number are low (Geodict, 2020).

$$\vec{u} = -\frac{K}{\mu}(\nabla p - \vec{f}) \tag{2.2}$$

The equation above demonstrates Darcy's law, which is used in FlowDict. In this formulation $\vec{u}$ is the fluid flow velocity, K is the permeability of the porous media, μ is the fluid viscosity, p is the pressure, f is a force density (Linden & Planas, 2019). In this formulation, it is assumed that the fluid is incompressible, meaning it has a constant density, and Newtonian, meaning it has a constant viscosity. Darcy's law only applies to Stokes flow, which is very slow and has a Reynolds number close to zero.

$$-\mu\Delta\vec{u} + \nabla p = f \tag{2.3}$$

The equation above is the Stokes conservation of momentum is a simplification of Navier-Stokes equations to describe the flow. In this simplification, the inertial term is dropped; therefore, the pressure drop and fluid velocity change linearly. Using appropriate solver to solve the partial differential equations describing the fluid flow, FlowDict predicts the porous media permeability as a tensor, including the permeability in different directions (Linden & Planas, 2019).

The choice of the physical equation to solve is also dependent on the particular material. When the material is made of only empty and solid voxels, Stokes or Navier-Stokes equations are used. Figure 6 demonstrates the choice of physical equations and solvers available in FlowDict.

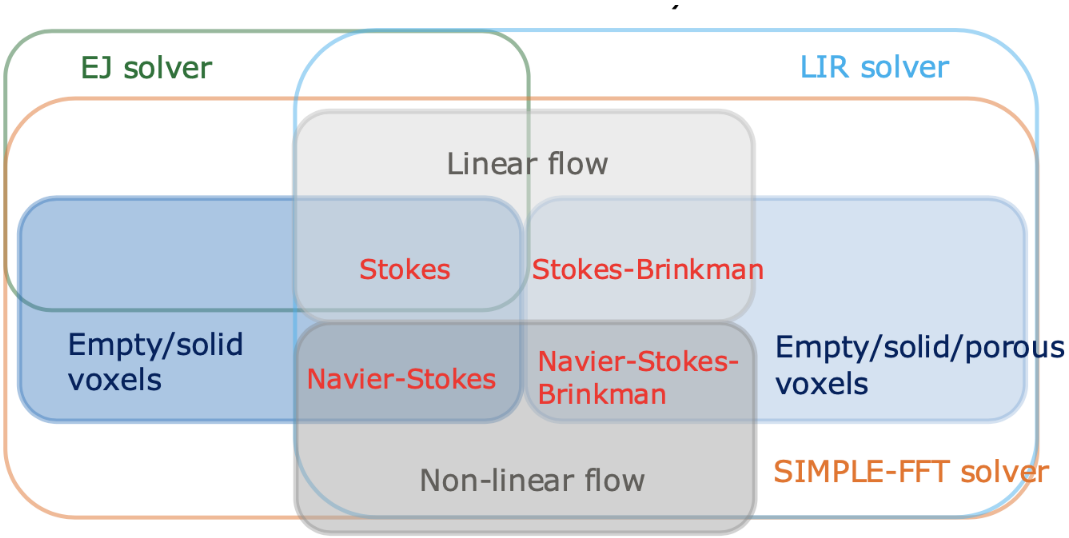


Figure 6: The choice of physical equations and solvers available in FlowDict (Linden & Planas, 2019)

Later on in this work, the results of pore network modeling using OpenPNM are compared with Geodict simulation results. This serves as a comparison between the pore network modeling approach and direct numerical simulations.

Chapter 3

Simulation Model Development

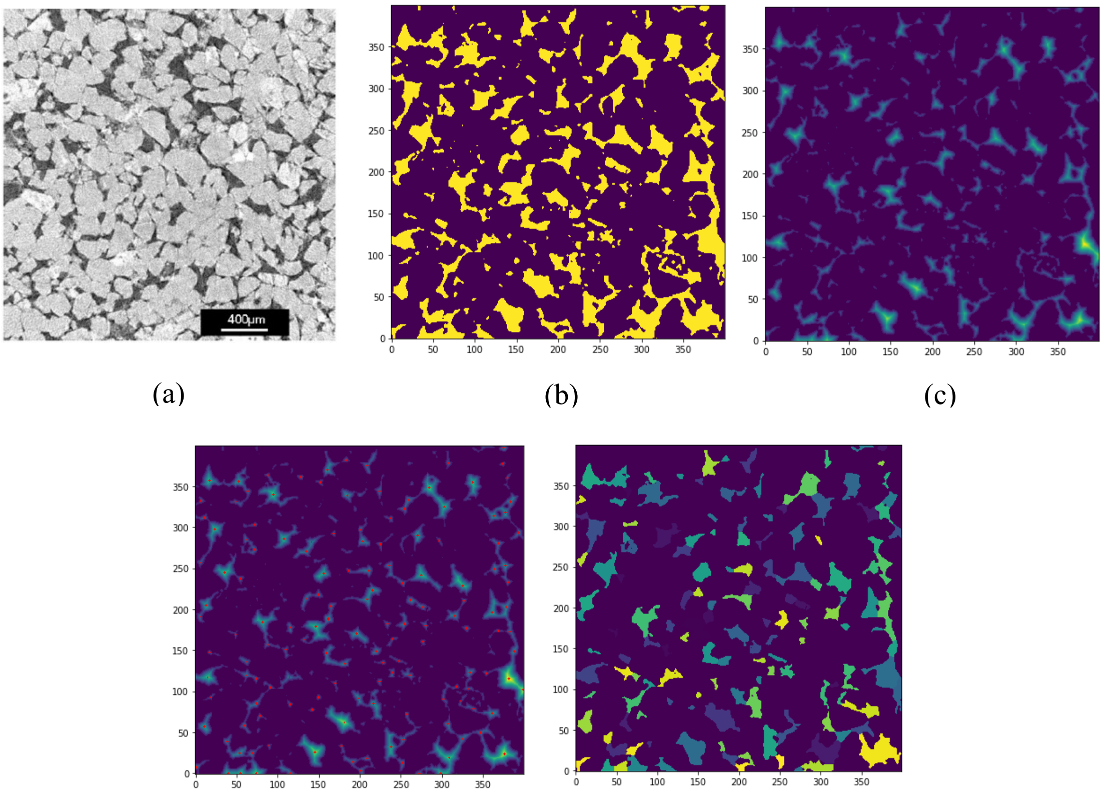
3.1 Pore Network Extraction Using Watershed Segmentation

To be able to perform transport simulations using pore network modeling, we need to have a stick and ball network, which represents the pore space of the rock. To derive this network of sticks and ball, we use segmented micro-CT images of a rock, in which voxels with a value of one indicate the rock matrix and the voxels with a value of zero indicate the pore space. Having these segmented micro-CT images, we use the SNOW algorithm to derive a stick and ball network, which represents the pore space, and then we can start performing flow simulations on this network.

The network extraction algorithm which is implemented in this work is presented by J. Gostick et al. (2017), which is an improved version of the watershed segmentation algorithm offered by Rabbani et al. (2014). They focused mostly on sandstone material, and the algorithm can provide many spurious peaks on the ridges and plateaus that can appear as local maximums in the distance transform map. J. Gostick et al. (2017) presented filters by which the over-segmentation of the image can be prevented. The algorithm is called a subnet of the over segmented watershed (SNOW) because it finds all the peaks in the beginning and then tries to reduce the spurious peaks by ruling them out. The algorithm is also implemented in a Python package named Porespy, which is used in the simulation of our models.

In the SNOW algorithm, the Euclidean distance transform map of the pore space is pre-filtered by a Gaussian blur filter to remove the plateaus, which are created in the case of two parallel solid walls. This Gaussian blur filter is represented by sigma, and the higher the sigma is, the lower the number of the peaks will be. Then the peaks at the distance map are identified using a spherical structuring element of R. If the R is small, then there will be a lot of spurious peaks on plateaus and ridges, and if the R is too big, then some information at small throats might be lost. In the end, peaks on the saddle points from the last steps are removed, and the nearby peaks are merged to eliminate the spurious peaks as much as possible (J. Gostick et al., 2017).

As demonstrated in Figure 7, after receiving the greyscale micro-CT image, the pore space is segmented from the matrix, which results in Figure 7 (b). This segmentation is out of the scope of this article, and the segmented image is the starting point of the SNOW network extraction algorithm. Then the distance transform algorithm is run in the pore space, which is indicated in the segmented image as in Figure 7 (c). In this distance transform, the brighter spots represent the pore space, which are further away from the matrix. To find the peaks of this distance transform, a spherical structuring element with a radius of R is then used, which can be specified during the network extraction. These peaks are defined by the red dots in Figure 7 (d). As can be seen, the points which are the brightest are chosen as the peaks since they have the largest distance to their surrounding rock matrix. Finally, the identified peaks are used as seeds for the watershed algorithm, and the pore space is separated into different regions. These regions are identified by random colors in Figure 7 (e) for illustration.



(a) (b) (c)

(d) (e)

Figure 7: Marker-based watershed segmentation of Berea sandstone (a) greyscale micro-CT image of Berea sandstone (Dong & Blunt, 2009) (b) Segmented image of Berea sandstone (please note that this segment does not correspond to (a)) (c) performing distance transform in the pore space (d) finding the peaks in the pore using a R max structural element (e) using watershed algorithm seeded with peaks to find the regions in the pore space

3.1.1 Image preprocessing

In a real image of a rock, The solid walls contain a lot of curvatures, which can cause problems for the distance transform algorithm, and it can result in many spurious peaks. Therefore, before running the distance transform J. Gostick et al. (2017) proposed to use a Gaussian blur filter to minimize these spurious peaks. This phenomenon might happen, e.g., in plateaus, where two parallel solid walls exist. This blur filter, which smooths the image, is represented by sigma in the extraction algorithm, and it is an adjustable parameter. The typical range of sigma is between 0 and 0.6, and as it increases, the number of local maximums in the distance map decreases. It is important to note that a large number of sigma can cause loss of data; therefore, an optimum number should be identified for sigma, to both prevent the over-segmentation and loss of data (J. Gostick et al., 2017). Later in this work, a workflow is presented to quality check for the optimum sigma by searching for the artifacts in the resulting primary drainage relative permeability curves.

3.1.2 Euclidean Distance Transform and Removing Spurious Peaks

In the Euclidean distance transform, the distance of a pore voxel is measured from the nearest voxel, which represents the rock. In case of a sphere pack as in Figure 5, the Euclidean distance of a pore voxel from the spheres can be calculated as:

$$Distance = \sqrt{(x_{voxel} - x_{sphere})^2 + (Y_{voxel} - Y_{sphere})^2 + (Z_{voxel} - Z_{shpere})^2} - R_{sphere} \tag{3.1}$$

This calculation can be computationally expensive and depends on the size of the image that it is being run on (Swanberg, 2012).

Figure 8 demonstrates the distance transform in a cross-section of Berea sandstone and a magnification of a crop. In this figure, the bright green colors represent the pore space with further distance from the rock matrix. The further the distance of pore space from the rock matrix, the green color is brighter.

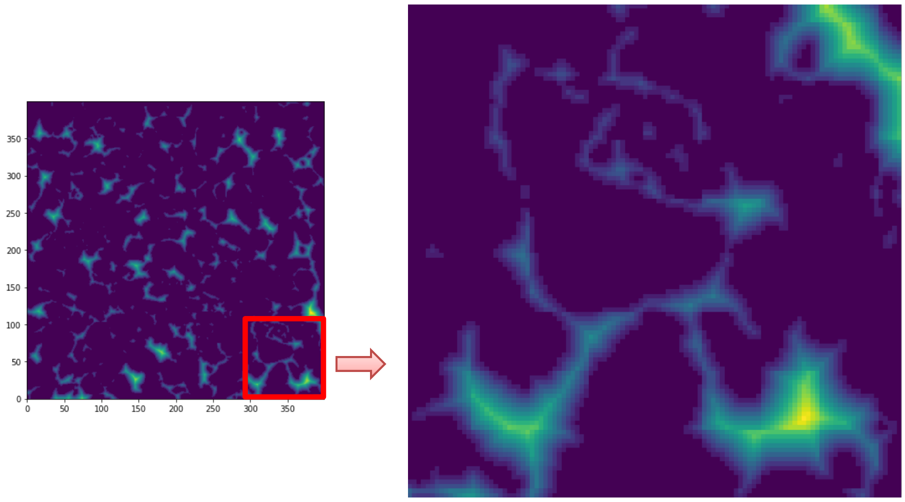


Figure 8: (Left)Euclidean distance transform demonstration in a cross section of Berea Sandstone (Right) magnification of a crop

In the Porespy package, which is used in this work, the spim function of the Scipy.ndimage Python package is implemented. In this algorithm, a maximum filter with a spherical structuring element R is used to define the maximums in the distance map. In the maximum filter, the local peaks in the neighborhood of each voxel are determined by the structuring element, and while the local peaks keep their value, the value in other voxels is replaced by the maximum value. The peaks are then identified by locating the voxels, which have an equal value to the distance map. In this filter, increasing the R will reduce the number of peaks in plateaus and ridges. However, similar to sigma, if R is too big, then some peaks are lost, especially in the pores that are smaller than the structuring element (J. Gostick et al., 2017).

Even with a proper choice of sigma and R, some spurious peaks can exist in the resulting peaks. That is why J. Gostick et al. (2017) presented two more filters to remove these spurious peaks. The first filter removes the peaks on the saddles, which are surrounded by several voxels with the same value in the distance transform map. These voxels can be connected to the voxel with higher values in the open pore space. The second filter merges nearby peaks, which is helpful in material with large and elongated pores, which causes over-segmentation of the large pore space. An illustration of how these filters work on a pore space is demonstrated in Figure 10.

3.1.2.1 Removing peaks on saddles

The aim of this filter is to remove peaks that are erroneously identified on the ridges since they are surrounded by several voxels which are having the same distance value, but they are connected to the open pore space with voxels with higher distance value. The algorithm proposed by J. Gostick et al. (2017) tackles this issue by using an iterative process. In the first step of this dilation, the peak is dilated with a cubical structuring element, and then it is flooded by the maximum value. After the flooding, the resulting map is compared to the previous distance map, and where the values are equal, new peaks are identified. Figure 9 (a) shows a 3D example of the saddle contour, and Figure 9 (b) demonstrates the corresponding distance map. Figure 9 (c-i)- (c-iii) demonstrates how the filter proceeds to find the erroneous points (J. Gostick et al., 2017).

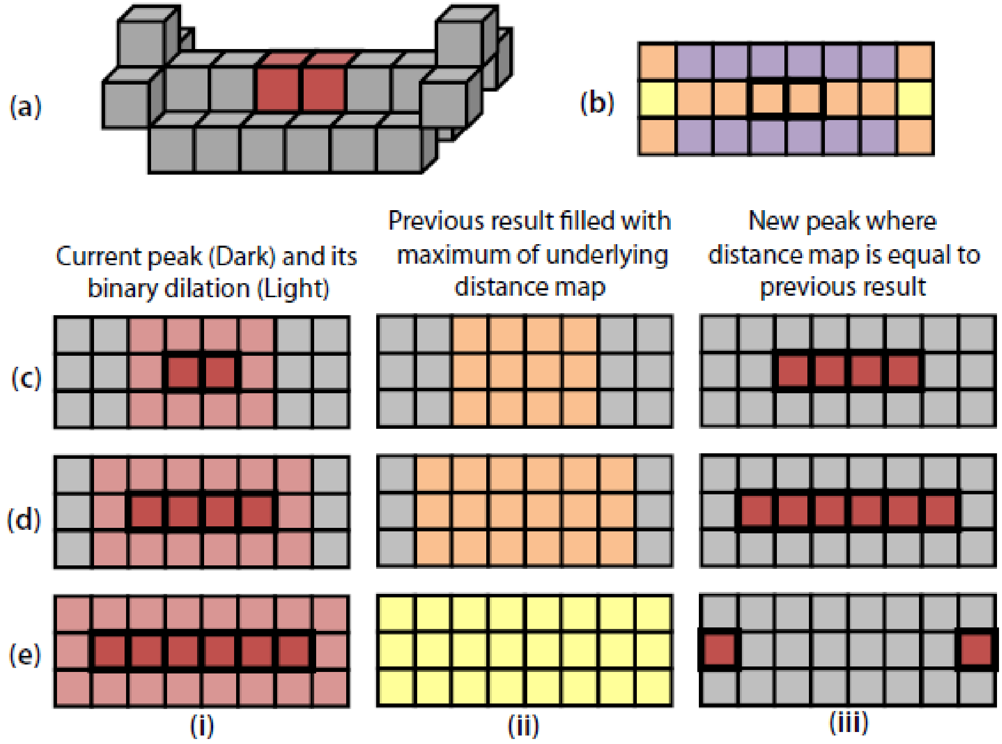

(a)
(b)
Current peak (Dark) and its binary dilation (Light)
Previous result filled with maximum of underlying distance map
New peak where distance map is equal to previous result
(c)
(d)
(e)
(i)
(ii)
(iii)

Figure 9: Removing peaks on saddles algorithm (J. Gostick et al., 2017)

3.1.2.2 Merging nearby peaks

As the existence of large pores in high porous materials can cause peaks that are too close to each other, this algorithm can help to identify and remove these peaks. This algorithm identifies these peaks by finding the distance between the peak and the distance of the peaks to the nearest solid surface. If a pair of peaks are found which are closer to each other than their distance from the solid surface, then the peak which is further away from solid is kept, and the other peak is removed (J. Gostick et al., 2017).

The process of removing spurious and erroneous peaks, which are initially identified by the watershed segmentation algorithm is demonstrated in Figure 10. In Figure 10 (a), we have the initial distance transform, and after using the R max element, the initial number of peaks are identified to be 203 in the corresponding cross-section of a sandstone (Figure 10 (b)). In Figure 10 (b), some of the spurious peaks are identified by red circles in the pictures, and these spurious peaks are then eliminated by using a Gaussian blur filter. Furthermore, the spurious peaks on the saddle points and the ridges, which are identified by the red circles in Figure 10 (c), are eliminated using trimming saddle points filter. Finally, the erroneous peaks, which are too close to each other, are identified and eliminated by merging nearby filters as demonstrated in Figure 10 (e).

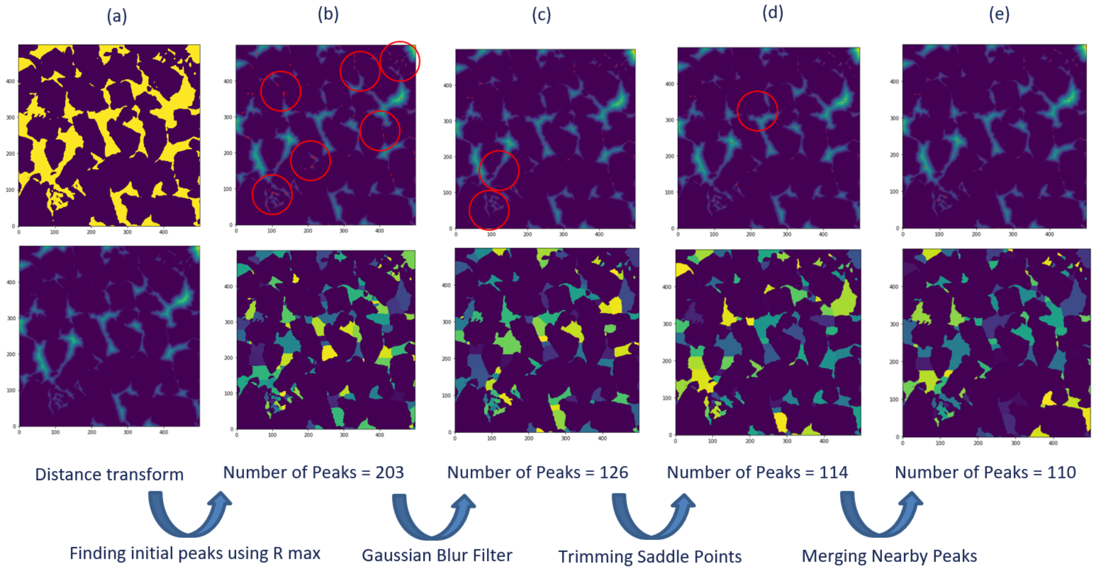


Figure 10: Illustration of filters existing in SNOW algorithm to remove spurious peaks of a sandstone rock. (a) the segmented micro-CT image of the sandstone and the corresponding distance transform map (the brighter spots represent the zones which have the largest distance from the rock matrix) (b) finding the initial number of peaks using structural spherical element R (c) usage of Gaussian blur filter to reduce the erroneous peaks (d) trimming saddle points filter to eliminate erroneous peaks on saddles and ridges (e) merging nearby peaks to eliminate the peaks that are too close together

3.1.3 Watershed Segmentation Algorithm

The watershed algorithm is a widely used method in image analysis used for image segmentation. It is based on hydrology in which it is used to divide the regions that water flows. In our context, the distance to the solid surface can be considered as depth in a topological map, and the largest distances, which are the pore centers are considered as downhills. In this analogy, when the water of a pore voxel flows towards a peak in the distance transform or a pore center, then it belongs to the region corresponding to that pore center (J. Blunt, 2017).

Now since we have the peaks of the distance transform identified by a structuring element of a sphere with a radius of R, we can insert these peaks into the watershed algorithm as markers. These markers will represent the pore centers or downhills, which will then result in segmented pore space. This process is demonstrated in Figure 11.

In Figure 11 (b), the distance transform is performed on the geometry of two connected pores, and the darker color represents the regions where the distance is maximum. After choosing the maximum distance as the peak and the pore center, a fluid is set to grow from the pore center (Figure 11 (c)) and then the first contact between these two growing fluids will mark the border between the pores which represents the throat between them (Baychev, 2019).

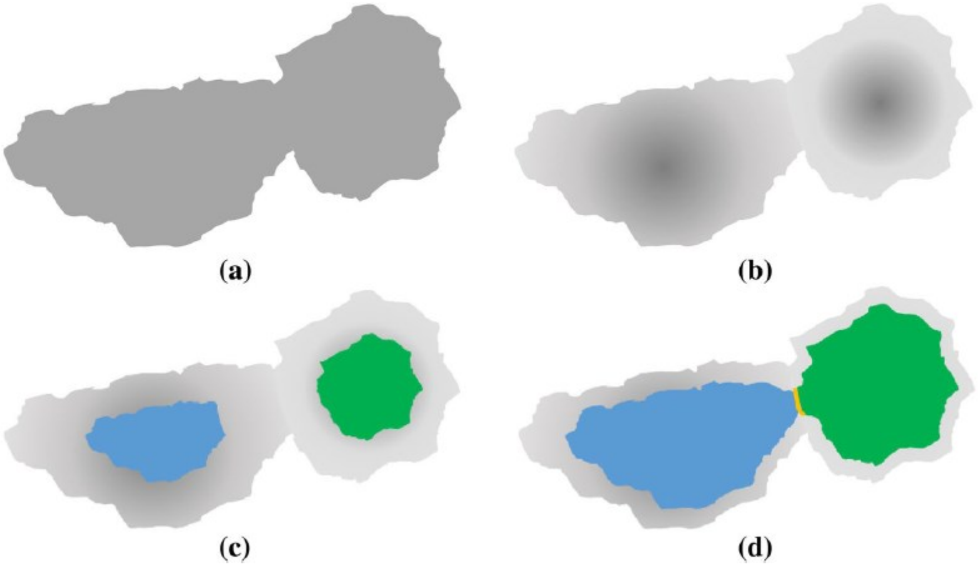


Figure 11: The illustration of the pore segmentation using the watershed algorithm. (a) geometry of two connected pores (b) distance transform perform on the pore space (c)growing fluid from pore centers (d) contact between the fluids causing the segmentation (Baychev, 2019)

After the fluid growth is complete, then the pore space is segmented into two regions, as demonstrated in Figure 12. The black dots in Figure 12 represent the center of the pores, and the boundary between them represents the connecting throat, which connects the two pores together.

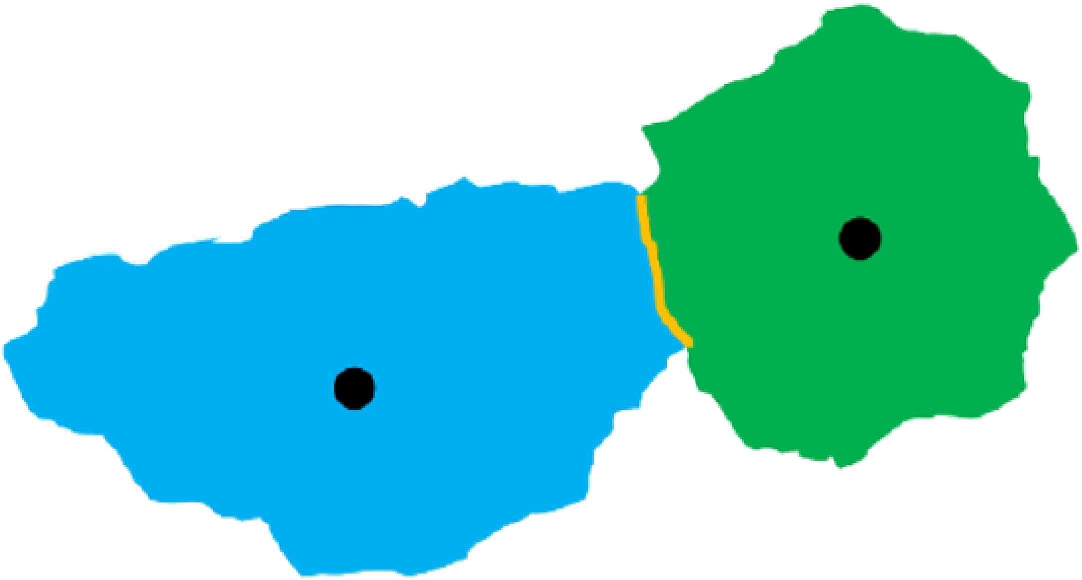


Figure 12: The pore centers which are identified as black dots and the boundary between the regions which represents the connecting throat (Baychev, 2019)

3.1.4 Obtaining Geometrical Parameters

After segmentation of the image into different regions, the geometrical properties are obtained using the fact that the pore centers are the maximums in the distance transform, and the throats are the space between these regions. A comprehensive explanation of the ways to obtain these geometrical parameters is presented in Porespy Python package documentation and *Versatile and efficient pore network extraction method using marker-based watershed segmentation* (J. Gostick et al., 2017).

In the context of the geometrical parameters, it is important to differentiate the inscribed and equivalent throat diameters. Inscribed throat diameters are identified from the maxima of the global distance transform. These diameters are the default output of the SNOW algorithm, and they are used to perform capillary pressure simulations. Equivalent throat diameters, on the other hand, are the diameter of a circle with the same area as the throat cross-section (J. Gostick et al., 2017). The equivalent throat diameters are used as pore throat diameters to simulate absolute permeabilities and relative permeabilities. These diameters are used to calculate conductivities, and as stated by J. Gostick (2017), results in better simulation predictions.

In Figure 13 it is demonstrated that after segmentation of the pore space of a sandstone, the network and regions 'information can be used to define the connectivity and size of the pores and throats. In Figure 13 (left), the image is segmented into different regions, and a crop of the image, which is identified by a red box, is magnified in Figure 13 (right). In this magnification, the pores are represented by spheres which are centered in the peaks of the distance transform map. The throats are represented by tubes which are connecting the spheres to each other. The colors in Figure 13 are randomly selected to distinguish different regions of the pore space.

In terms of geometrical properties, it should be noted that there are two approaches to calculate the pore volumes. In the first approach pore volumes are the summations of the volumes of voxels, which are included in a region corresponding to a pore; however, these pores in most of the cases do not have a spherical shape.

In the second approach to have a network model consisting of balls and sticks, the geometrical properties which are extracted from SNOW might not be representative of our model. This means after the pore network is extracted using SNOW, some parameters must be recalculated to represent the desired model. In this approach, since a network model of balls and sticks is implemented, pore areas and throat conduit lengths are recalculated after the extraction to match the assumptions of a pore network with balls and sticks. This recalculation involves spherical pore area and pore volumes based on pore diameters. The choice of which approach to use for the pore volume calculations will affect the input pore late filling parameters which will be elaborated in the sensitivity analysis part of the Berea sandstone.

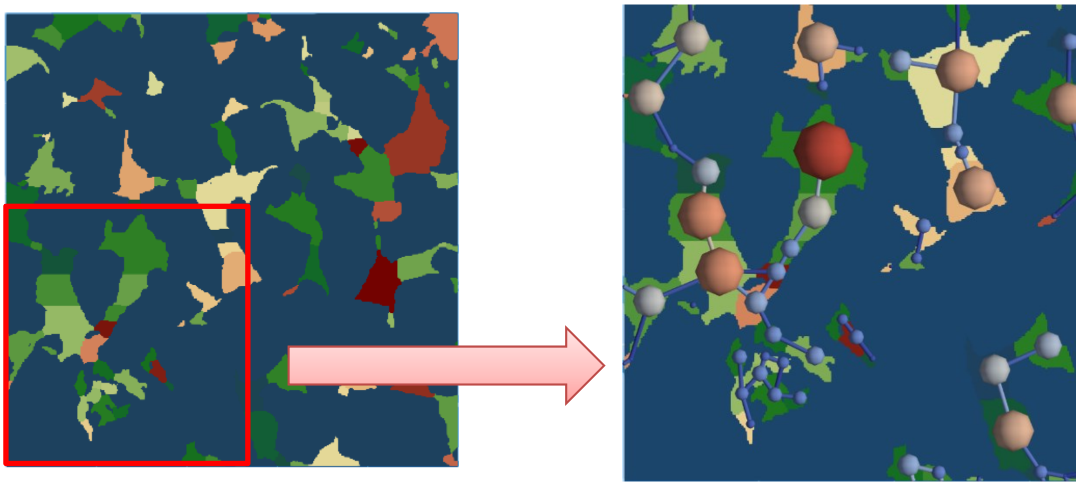


Figure 13: (Left) Segmentation of the pore space of a sandstone into different regions using SNOW (Right) illustration on obtaining geometrical parameters

3.2 Transport Predictions

After obtaining a PNM consisting of sticks and balls, the algorithms based on the displacement physics are assigned to the network to predict petrophysical rock properties. In this section, the formulation used to derive the transport properties in our simulations will be explained. These properties include primary drainage capillary pressure, absolute permeability, and primary drainage relative permeability.

After extracting a pore network from a micro-CT image of a rock and deriving the corresponding geometrical properties, we want to derive the intrusion curve of the network and define the invasion state at each capillary pressure step. This is done using the percolation theory and the capillary pressure of the pores and throats of the network. It is important to note that, since in this thesis, we are only simulating the primary drainage process, the capillary pressures of the throats are the limiting factor, and the pore capillary pressure is set to zero in each simulation.

The Young-Laplace formulation and the pore network from the SNOW extraction algorithm are used to calculate the capillary pressure of the throats in the network, as demonstrated further in this section. During the simulations, it was evident that using solely the pore network geometrical parameters extracted by the SNOW algorithm; the capillary pressure of the experimental data can not be matched. This is interpreted to be due to the resolution limitations of the micro-CT images, and the ball and stick geometrical simplification. To compensate for this limitation and simplification, a pore late filling model is implemented in the simulations of capillary pressure curves, which is also further discussed in the section.

3.2.1 Capillary Pressure

3.2.1.1 Capillary Pressure Young-Laplace Equation

In our pore network modeling, we are assuming a pore network connection of cylindrical tubes. Therefore, to calculate the capillary pressure, we can use the analytical relationship derived by Young-Laplace for capillary pressure as (J. Blunt, 2017):

$$P_c = -\frac{2\sigma\cos(\theta)}{r} \tag{3.2}$$

In this equation, r is the pore throat radius, P_c is the capillary pressure, σ is the interfacial tension, and θ is the contact angle between two immiscible phases and a solid surface. Typically, the contact angle is measured from the denser phase, as demonstrated in Figure 14. When two immiscible fluids are in contact with a solid surface, one fluid is usually attracted more strongly than the other fluid. The phase which is more strongly attracted is called the wetting phase. This definition further implies that in case of having a contact angle less than 90 degrees, the denser phase is the wetting phase. Furthermore, having a contact angle of 90 degrees means that the system is intermediate or neutral wet, and in the case of having a contact angle of more than 90 degrees, the less dense phase is the wetting phase (Tranter et al., 2007).

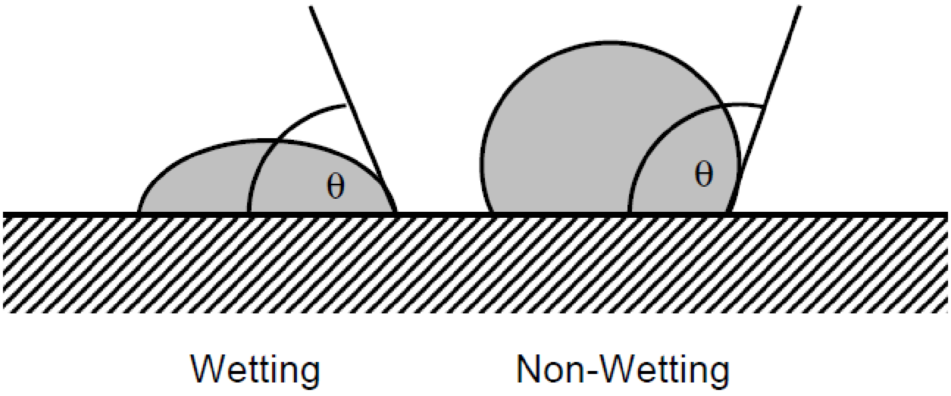


Figure 14: Demonstration of contact angle measurement and the effect on wettability (Gostick J. , 2008)

3.2.1.2 Late Pore Filling

As stated above, the Young-Laplace equation alone is inadequate to match the simulated capillary pressure curve with the experimental data. This mismatch is due to the limitation in the micro-CT image resolution and the simplified ball and sticks geometry, which is used in pore network modeling. Therefore, a late pore filling modeling is proposed to compensate for the limitation.

The pore late filling model is based on the concept of fractal geometry. In this concept, the fractal dimensions are the degree from which a curve, a surface, or a volume is different from its ideal topological shape (Y.-H. Lee et al., 1990). From the definition of a fractal surface in which the dimensions are fractal rather than integer we can have (Li, 2010):

$$N(r) \propto r^{-D_f} \tag{3.3}$$

In this formulation, r is the radius of the unit, which is chosen to fill the fractal object, N(r) is the number of the units with the radius r that can be filled in the object, and D_f is the fractal dimensions. Furthermore, according to the capillary tube model, we can have (Li, 2010):

$$N(r) = \frac{V_{Hg}}{\pi r^2 l} \tag{3.4}$$

In which l is the length of the capillary tube, r is the radius of the capillary tube, and V_{Hg} is the cumulative volume of mercury, which has intruded the rock. Substitution of equation (3.4) into equation (3.3) yields:

$$\frac{V_{Hg}}{\pi r^2 l} \propto r^{-D_f} \tag{3.5}$$

And then we can reduce equation (3.5) to have:

$$V_{HG} \propto r^{2-D_f} \qquad 3.6)$$

Substituting equation (3.6) into (3.2) results:

$$V_{HG} \propto P_c{}^{-(2-D_f)} \qquad (3.7)$$

In a real porous media of a rock, it is assumed that the rock interface is fractal, and it is so complex that an angular geometrical shape cannot describe it. In the pore space of a rock, the pores and throats extend over a wide range of lengths because of the limited resolution of the imaging techniques and inability to capture the complex geometry. (Tsakiroglou, , 1993) proposed a fractal model to compensate for this phenomenon (Jeff T. Gostick et al., 2007; Chang et al., 2002). Furthermore, (Chang et al., 2002) proposed a model to capture the pore structure, which is not resolved in the pore network as below:

$$\frac{S_{w,i}}{S^*{}_w} = (\frac{P^*{}_{c,i}}{P_c})^{\eta} \qquad (3.8)$$

In this formula, the $P^*{}_{c,i}$ and $S^*{}_w$ are the capillary pressure and wetting phase saturation of pore i, upon the first invasion, and $S_{w,i}$ is the residual saturation of the wetting phase in the pore i. This means that when the nonwetting fluid is invading into the pore i at a capillary pressure of $P^*{}_{c,i}$, the remaining fluid occupies a fraction of the pore volume, which is denoted as $S^*{}_w$. $S^*{}_w$ is the fraction of the pore space missing from smaller length scales, which is not represented in the network model due to micro-CT image resolution and PNM geometrical simplifications. Therefore, some of the pore space is missing the measured throat or pore size distribution, or because the measured volume of each pore is assigned to a regular geometric object. $S^*{}_w$ is parameter that we need to input in our modeling to calculate the pore late filling parameters.

In this formulation, η is the fitting parameter that follows the fractal dimensions of the pore scale. In the case of the nonwetting phase drainage at $P_{c,i}$ > $P^*{}_{c,i}$, it stands between zero and one for a fractal dimension of two to three. Following the derivation of (Sahouli et al., 1999) based on the definition of a fractal surface:

$$\eta = D_s - 2 \qquad (3.9)$$

In this formula, D_s is the fractional dimension of the pore space. The fractal dimensions of the rock pore can be computed from different methods based on the microtomography images of the rocks, e.g., using the FracLac plugin of ImageJ image processing software. (Peng et al., 2011) used a box-counting method to identify the fractal dimensions of different rocks from grayscale CT images. The fractal dimensions proved to increase by increasing the porosity and complexity of the structure of the pore space (Peng et al., 2011). It is also important to note that this fractal dimension of the rock is also dependent on the resolution of the image (Huang et at., 2019). (Krohn, 1988) identified the fractal dimensions of different sandstones by plotting the logarithm of the number of features (N(r)) versus the logarithm of the feature sizes (r), coming from SEM images of rocks with different magnifications and therefore different resolutions.

In the modeling of this work, throats volumes are set to zero, and late pore filling is only implemented on the pore volumes. To find the $P^*{}_{c,i}$ of the pores in the network, $P^*{}_{c,i}$ of the pores are adapted from the capillary pressure of the throats, which are connected to the pore. This adaptation is made in three different ways, which are also available in the OpenPNM package. $P^*{}_{c,i}$ of a pore can be either the minimum, maximum, or arithmetic mean of the capillary pressures of the throats, which are connected to a pore. The sensitivity of this mode selection on capillary pressure and relative permeability curves are later presented in this work in Chapter 4.

3.2.1.3 Drainage Simulations

Since throats are always smaller than the pores, in the drainage process, the entering of throats requires higher capillary pressure. Therefore, after calculating the capillary pressures of the pore throats in the network, an algorithm is needed to perform a proper drainage simulation. This algorithm decides how the invading phase enters the pore network.

One option to simulate the drainage simulations are using the quasi-static invasion percolation theory, which is already programmed in the OpenPNM package (Gostick J. A., 2016). In this algorithm, the path which has the least resistance to flow is identified by ranking the pore throats in front of the flow. Then, the invading phase invades the throats, which have the least resistance to the invasion, i.e., the throats with the least capillary pressure. On the other hand, the capillary pressure in the intrusion curve is defined by the highest capillary pressure that the invasion front has invaded. To explain in more detail, the algorithm stores all the accessible throats and then try to search for the throat with the least capillary pressure to invade next. This process can have a more accurate representation of the flow relative to ordinary percolation theory. However, the problem is that this algorithm is more computationally expensive and takes more time (Gostick J. A., 2016). The results will be the saturation of the nonwetting invading phase versus the pressure applied to the network. In Figure 15, this algorithm is demonstrated at different invasion sequences. The inlets are set at the left side of the cubical network, and the invasion sequence in increasing from left to right. Furthermore, throughout this work, unless otherwise stated, all the drainage simulations are conducted along the X-axis from left to right.

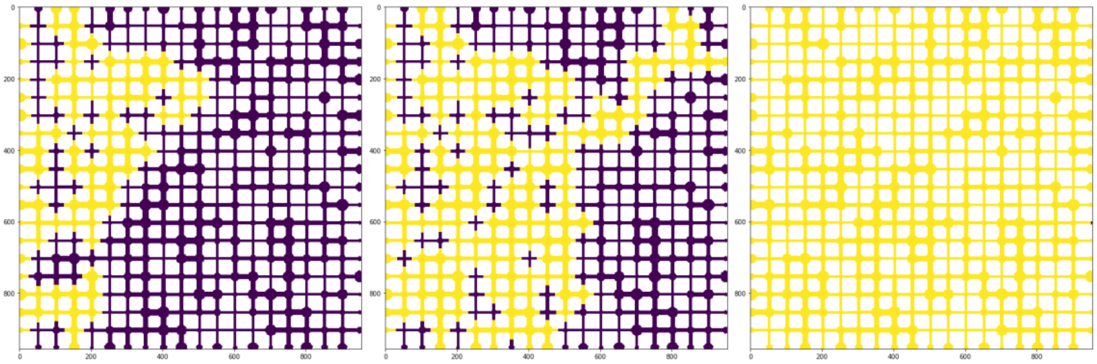


Figure 15: Invasion percolation algorithm demonstrated at an increasing invasion sequence from left to right in a synthetic cubical network

A trapping mechanism could also be implemented into the invasion percolation algorithm with which one can model the hydraulic trapping of the wetting phase after the invasion of the nonwetting phase into the network. In this trapping mechanism, the defending phase can be trapped when it is surrounded by the invading phase, which means that the fluid in those pores is bypassed, and saturation of the non-wetting phase will not reach one. In the OpenPNM package, this mechanism is implemented by Gostick J. A. (2016) based on the algorithm proposed by (Masson, 2016).

The intrusion curve, which is the result of applying the trapping mechanism on a synthetic pore network, is demonstrated in Figure 16. In this figure, the difference with the case in which the trapping mechanism is not implemented is evident, since the intrusion curve is not reaching a nonwetting phase saturation of one. In Figure 16, there are some minima observed in the intrusion curve. The reason for this behavior is the algorithm behind invasion percolation. At each step of the percolation, the algorithm might face throats that have capillary pressures less than the ones observed in the previous steps.

For the invasion percolation shown here in Figure 16 the inlet for the invasion of the invading phase is set at the left side of the network and the outlets are set at the right side of the generic network.

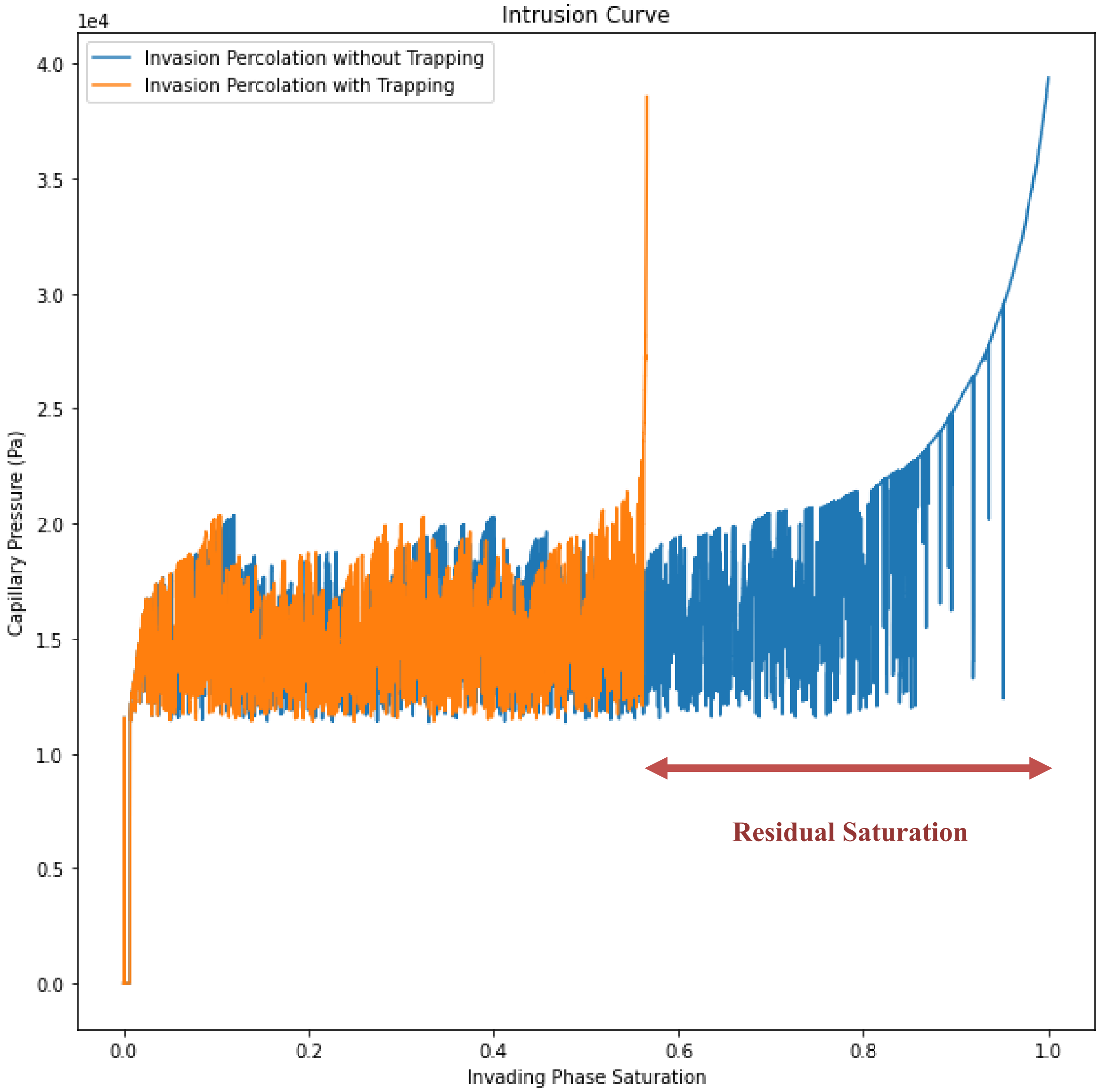


Figure 16: Intrusion curve in case of trapping mechanism implementation curve on a synthetic cubical network

3.2.1.4 Mixed Invasion Percolation

Another percolation algorithm that exists in the OpenPNM package is the mixed invasion percolation algorithm. Mixed invasion percolation is a special case of invasion percolation algorithm in which the pores and throats can be invaded on an individual basis. In the case of running the mixed invasion algorithm in the bond mode, the entry pressure of the throats is used and the connected pores are invaded automatically.

In a pore network, throats are typically smaller than the pores and for the drainage process, the throats must be entered with a higher capillary pressure. Therefore, in this work to simulate the drainage process, the mixed invasion percolation algorithm is used in the bond mode. To do so the pore entry pressures are set to zero so that the connected pores to the throats are invaded automatically. This convention can speed up the percolation algorithm since the throats are typically smaller than the pores.

Furthermore, there are more options implemented in the mixed invasion percolation algorithm, and it is more flexible regarding the percolation results. In the mixed invasion percolation algorithm with the increase of the saturation of the invading phase, the capillary pressure is always increasing. The intrusion curve for the mixed invasion percolation algorithm is demonstrated in Figure 17. In this figure, it can be seen that there is a difference in the residual trapping resulting from the mixed invasion percolation algorithm and the invasion percolation algorithm due to the algorithms behind the two different percolations.

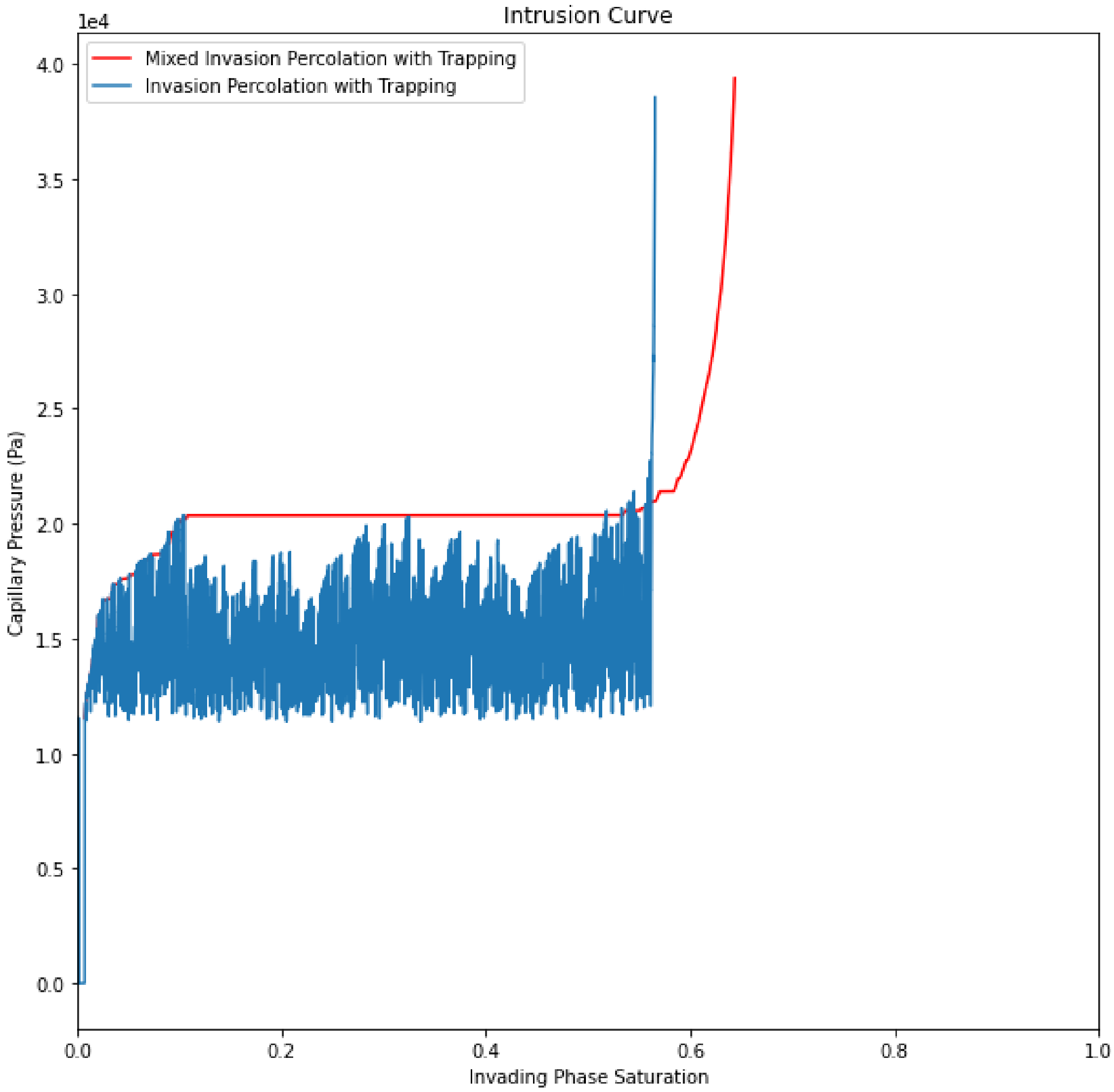


Figure 17: Intrusion curve demonstration for mixed invasion percolation

3.2.2 Absolute Permeability

Absolute permeability is an important physical property of a rock, which is defined as the ability of a rock to pass a fluid through its interconnected pores. This property can easily be derived from a pore network model. While absolute permeability is a measure of the conductance of a porous medium when pores are a 100% saturated with one fluid, effective permeability is a measure of the conductance when the medium is saturated with more than one fluid. The relative permeability is defined as the ratio of the effective permeability of one fluid in presence of another over the absolute permeability.

After setting constant pressure boundary conditions on the inlet and the outlet of the pore network, the absolute permeability is calculated by solving for the flow rate, which is caused by the set pressure difference. To do so, we solve a steady-state transport problem by solving a linear series of equations. These series of equations consist of the mass balance for each pore in the network. Having the pressures of the pores inside the pore network, we can derive the flow rate of the fluid traveling inside the network from Hagen Poiseuille conductivities. Finally, using Darcy's law, we can derive the absolute permeability of the pore network.

After setting constant pressure boundary conditions on both ends of the pore network, the solver identifies the pressure of each pore in the pore network. Having the pressure of each pore in the network, the flow which is happening at one inlet of the pore network is calculated using equation (3.10). This flow rate is due to the pressure difference at the boundary of the network and the pores which are connected to this boundary. Absolute permeability is then calculated using Darcy's law using the calculated flow rate and the constant pressure boundary conditions.

The mass balance equation which is solved over each pore of the network is as (Jeff T. Gostick et al., 2007):

$$q_i = \sum_{j=1}^{n} g_{h,ij}\left(P_j - P_i\right) = 0 \tag{3.10}$$

Where i and j are the pore index and q stands for the net flow through the pore, n is the total number of the pores, g is hydraulic conductance of the throat, which is connecting the pore i to pore j, and P is the pore pressure. The hydraulic conductivity of the throats is defined based on Hagen Poiseuille formulation:

$$g = \frac{A^2 * shape\ Factor}{8\pi\mu L} \tag{3.11}$$

Where A is the throat area, μ is the fluid viscosity, and L is the conduit throat length, and in case of a pore, it is the half pore length. The shape factor is derived from (Akbari et al., 2011) formulation to calculate the conduit shape factor for a network with spherical balls and sticks as throats. We use this shape factor to account for the flow, which is happening in a network consisting of sticks and ball.

$$Shape\ Factor = \frac{L}{A^2} / (\frac{4}{D^3\pi^2}\left(\frac{2DL}{D^2 - 4L^2} + tan^{-1}\left(\frac{2L}{D}\right)\right)) \quad (3.12)$$

In the shape factor formulation, L stands for throat conduit length, A is the pore or throat area, and D is the pore or throat diameter (Akbari et al., 2011). The conductance of a pore throat which connects a pore i to a pore j is calculated by a harmonic mean including shape factors as:

$$\frac{1}{g_{h,ij}} = \frac{SF_{pi}}{g_{h,pi}} + \frac{SF_t}{g_{h,t}} + \frac{SF_{pj}}{g_{h,pj}} \quad (3.13)$$

In this formulation $g_{h,ij}$ is the conductance of the throat connecting the pore i to pore j, SF_{pi} is the shape factor for pore i, and $g_{h,pi}$ is the conductance of pore. The other variables in the formula correspond to the same parameters while the indices, t stands for a throat, and p_j is pore j. Figure 18 demonstrates the kind of setting that the calculation of the throat hydraulic conductance applies to.

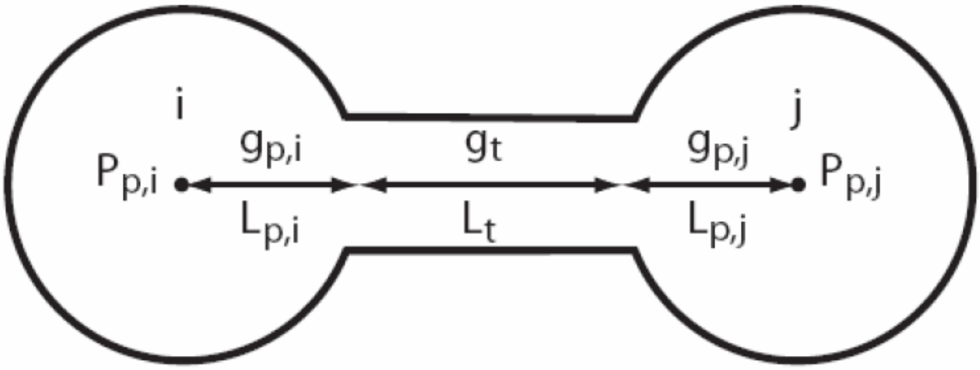


Figure 18: Conductance of a throat connecting pore i to pore j (Valvatne, 2004)

Equation (3.10) is solved using the steady-state transport solver for the Stokes flow in the OpenPNM package. In this algorithm, after defining the fluid properties, hydraulic conductance, and the inlet and outlet boundary conditions, the solver calculates in the pore pressure and the flow rate of the network. After obtaining the volumetric flow rate at the inlet of the sample, absolute permeability can be calculated using Darcy's law:

$$Q = \frac{KA}{\mu L}(P_{in} - P_{out}) \quad (3.14)$$

Q is the volumetric flow rate in the inlet, K is absolute permeability, μ is fluid viscosity, L is the porous medium length, P_{in} is the pressure at the inlet boundary, P_{out} is the pressure at the outlet boundary, and A is the flow area available to flow. The Area available to flow can be derived for a cubical microtomography image as:

$$A = (resolution * voxel\ size)^2 \tag{3.15}$$

It is important to note that this can change regarding the direction of flow. In this work, unless otherwise stated, the flow is happening along the X-axis from left to right. Therefore, L is the sample length in the X direction, and A is the area of the sample in the YZ plane.

3.2.3 Relative Permeability

Since we have a multiphase system, to derive the petrophysical properties of a rock we need to acquire the relative permeability with PNM. In our simulations, primary drainage relative permeabilities of pore networks are calculated and then compared with experimental data to check the accuracy of the results.

Relative permeability is calculated by modifying the conductivity of the individual throats that are invaded by the invading phase using invasion percolation theory. To determine the conductance of the throat conduits after the partial invasion of the network with a nonwetting phase, the conduits which are not available to a phase are closed. This closure means that the conductance of these throats is multiplied by a factor of 10^{-6}. The reason that the throat conductances are not multiplied to zero is to prevent solver errors. This can be done in different modes, which define how aggressive the closure will be. There are three different modes included in the OpenPNM python package. First is ‘strict’ mode in which any pore or throat, which is unoccupied by the phase, is closed. The second mode is ‘medium’, which closes the conduit in case either the throat or pores are unoccupied by the corresponding phase. In the third mode, which is named ‘loose’ the conduit is closed only if the throats are unoccupied. Throughout this work, ‘medium’ mode is implemented for the calculations since it demonstrated to match the experimental results of the validation samples more accurately.

After the network is partially invaded by the nonwetting phase, and the conductivities are recalculated, the effective permeability of each phase is calculated using the Stokes flow algorithm. Therefore, this effective permeability is then divided by the single-phase absolute permeability using equation (3.16). The calculation of this effective permeability is done in the same way as for absolute permeability. However, here we are using the recalculated conductivities of the throats. Therefore, relative permeability can be obtained from the formula below:

$$K_{r,p}(s_p) = K / K_{eff,p}(s_p) \tag{3.16}$$

In this equation, K is absolute permeability, $K_{eff,p}$ (s_p) is the effective permeability of phase p at phase saturation of s_p and $K_{r,p}$ (s_p) is the relative permeability of phase p at phase saturation of s_p (Jeff T. Gostick et al., 2007).

It is important to note that, as it is shown in the model validation chapter later in this article, using this formulation to derive relative permeabilities, results in erroneous results for the relative permeability before the breakthrough. The reason for this erroneous data is that, before the nonwetting phase breakthrough, the invading phase has not formed a connected pathway through the sample yet, which will result in extremely low relative permeabilities.

Chapter 4

Model Validation and Sensitivity Analysis

For validating the models, segmented microtomography images of Berea and Bentheimer sandstones are used and the simulation results are compared with existing experimental data.

4.1 Berea Sandstone

Berea sandstone is a widely used material to study flow in porous media. It is found in Berea, Ohio, in Michigan basin and contains minor amounts of feldspar, dolomite, and clays. Here the upper unit of Berea sandstone is used since it is frequently used in core flooding experiments (Oak, 1990; Dong & Blunt, 2009), and there is plenty of data available for the rock sample to validate the models against which include absolute and relative permeability data (Oak, 1990) (Dong & Blunt, 2009). Moreover, Churcher et. al (1991) investigated the rock properties of Berea sandstone, including its capillary pressure, which we also used to validate our capillary pressure models.

The microtomography image is a subsection with a porosity of 19.6% and a size of $400^{3,}$ which is obtained from the imperial college website and used in other studies as well (London, n.d.; Dong & Blunt, 2009). The image resolution is 5.345 µm, and a cross-section of the micro CT image is demonstrated in Figure 19.

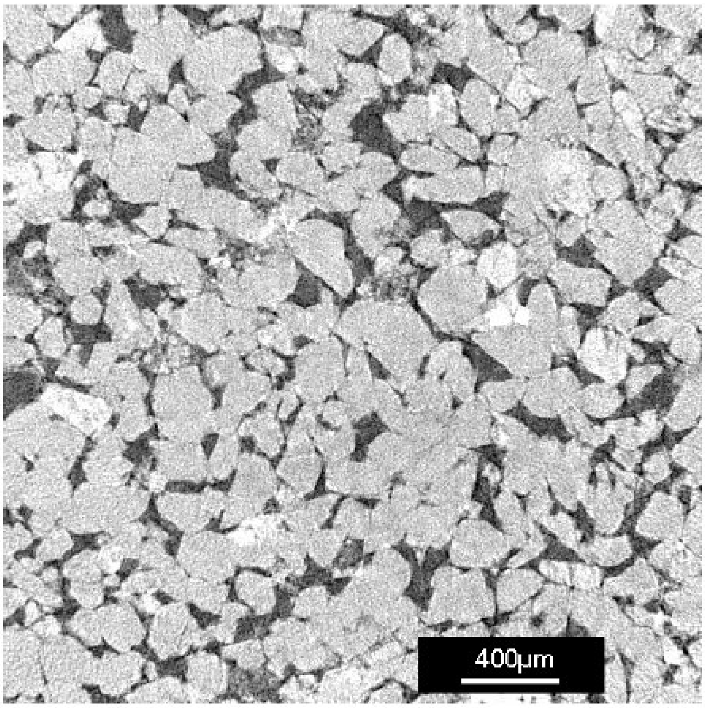


Figure 19: A cross-section of micro CT image of Berea Sandstone obtained from Imperial College website (Dong & Blunt, 2009)

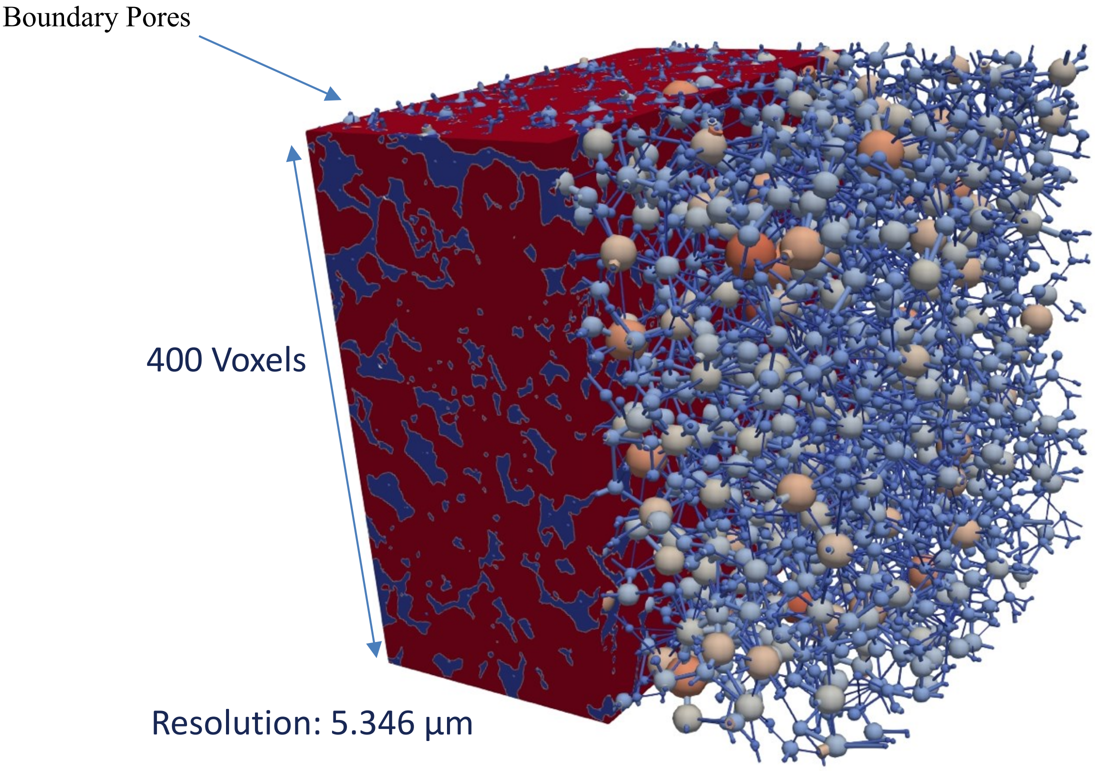


Figure 20: Pore network for Berea sandstone which is extracted using SNOW algorithm

The extracted pore network from the micro CT image is shown in Figure 20. In this figure, the pores and throats are scaled and colored by the inscribed pore diameter and inscribed throat diameters, respectively. The difference between the colors of the pores are more evident than the throats since the pore diameter variations are higher. The blue space in the micro CT image represents the pore space, and the red space represents the solid phase. There are some small pores and throats around the network which account for the boundary pores. These pores are used to assign boundary conditions on the network during transport simulations. The boundary pores are created outside of the micro-CT image by the SNOW algorithm.

Absolute permeability and residual saturation results of Berea sandstone are shown in Table 1. The input parameters to extract the network and run the simulations are demonstrated in Table 2. The fluid properties which are used in the simulations are illustrated in

Table 3. The fluid properties are based on the experiment carried out by Churcher et. al (1991) and Oak (1990).

SNOW extraction parameters, which are shown in Table 2, are based on the workflow, which is later presented in this work in the discussion section. The optimum Sigma and R_max are selected based on simulation of relative permeability curves from networks with different sets of sigma and R_max and choosing the most appropriate curve. Then, the relative permeability curve with the least artifacts and the most reasonable shape is selected. Pore entry pressure is set to be zero since we are simulating primary relative permeability drainage using the mixed invasion percolation in the bond mode and the throats are providing the greatest resistance to flow. Throat volumes are also set to zero to neglect the error caused by them. The η parameter of 0.85 for the late pore filling model, as stated in equation (3.8), is taken from Krohn (1988). Later in this chapter, a sensitivity analysis is conducted by changing η in a range of 0.1 to see the changes of capillary pressure and relative permeability curves with respect to the parameter. P_c^* mode changes, which are the pore capillary pressure adaptations from throat capillary pressures, are also investigated. Here for the Berea sandstone simulations, it has been set to medium to match the experimental results.

The S_w^* parameter of the pore late filling for Berea sandstone is obtained from Figure 21. The numbers indicated in this figure are derived from equation (4.1). Since the Berea sandstone image has a resolution of 5.345 μm, assuming a mercury surface tension of 0.48 N/m and a contact angle of 140 degrees, the highest values of capillary pressure the microtomography image shows is 2.75 bars. The connection between the capillary pressure and the image resolution lies within the fact that a capillary pressure represents a pore throat diameter, according to the Young-Laplace formulation. Therefore, as demonstrated by the red lines illustrated in Figure 21, we are missing on 21.4 percentage of the pore volume due to resolution limitation. The calculation of 2.75 bars capillary pressure is shown in the equation below:

$$2.75 * 1E5 = -\frac{2 * 0.48\, cos\,(140)}{(\mathbf{5.345} * 1E - 6)/2} \tag{4.1}$$

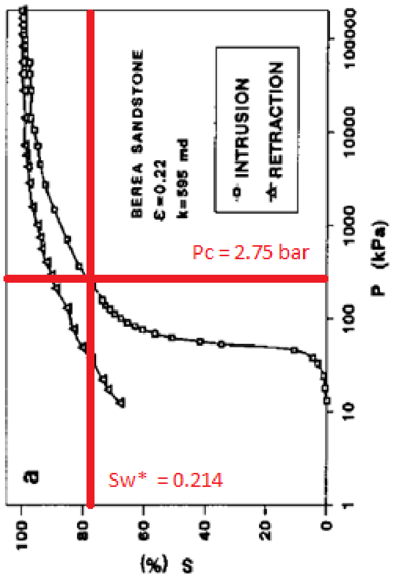


Figure 21: Experimental capillary pressure curve of Berea sandstone (Tsakiroglou et al., 1993)

Table 1: Prediction results of Berea sandstone

Parameter	**Value**
Permeability from Lattice Boltzman (J. Gostick et al., 2017)	1280 mD
Permeability from SNOW	1144 mD
Permeability from Geodict	1877 mD
Experimental Permeability (Dong & Blunt, 2009)	650 mD
Resolution	5.345 µm
Porosity	0.196
SNOW residual saturation from trapping mechanism	0.243
Experimental primary drainage residual saturation	0.24
Average Coordination Number	3.7

As the absolute permeability results are shown in Table 1, the SNOW algorithm resulted in the permeability of 1144 mD, which is good agreement with the results of Lattice Boltzman predictions (J. Gostick et al., 2017). However, it is still double the experimental absolute permeability (Dong & Blunt, 2009).

Here it is important to note that, as recommended by J. Gostick et al. (2017), equivalent throat diameters are used to derive the values of absolute permeability from SNOW simulations. The reason for this choice is to have a closer match to the Lattice Boltzman simulation results. Also in our simulations, using equivalent throats diameters were necessary to acquire acceptable relative permeability simulation results. Therefore, through our simulations we stick to the convention of using equivalent throat diameters for the absolute and relative permeability simulations.

Using equivalent throat diameter means to use the diameter of a circle with the same area as the throat cross-section. The default settings in the Porespy package uses inscribe throat diameters, which uses the maximum of global distance transform to define the throat diameters. In this context, it is essential to mention that to simulate capillary pressures, the inscribed throat diameters are implemented, whereas to simulate relative permeability curves, equivalent throat diameters are also used with the reasoning mention before.

The results of Geodict pore network software are also demonstrated here, which is less accurate than the SNOW prediction; however, this comparison is out of the scope of this article. One is for this loss of accuracy is the Geodict predictions were run using just the basic settings, and these predictions have room for improvement.

The trapping mechanism implemented in OpenPNM results in a residual network saturation of 0.243, which is in good agreement with the experimental residual saturation of primary drainage presented by Oak (1990). The average coordination number of the sample is 3.7, and the histogram of the pore coordination numbers is shown in Figure 22. Here again, It is important to note that for this comparison, the Geodict simulation was run on basic settings without the required modifications, which explains its deviations from the experimental results.

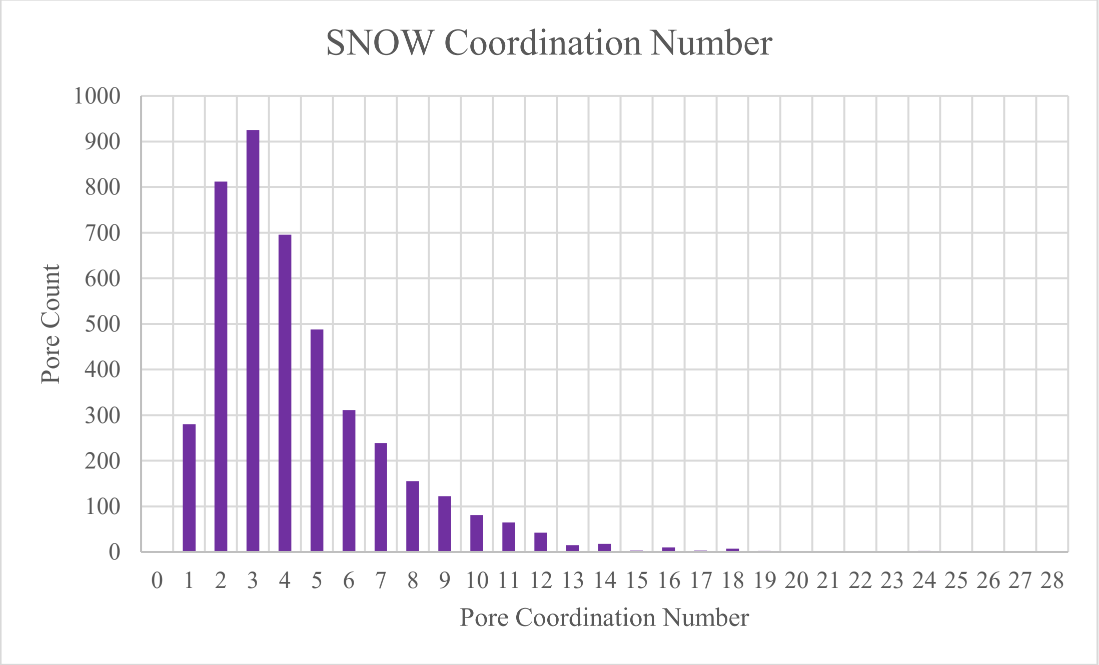


Figure 22: Pore coordination number histogram of Berea Sandstone

The results of the capillary pressure simulation of the SNOW algorithm are shown in Figure 23, along with the comparison with experimental and Geodict results. Please note that in this simulation, the trapping mechanism is activated for both capillary pressure and relative permeability simulations. The experimental capillary pressure curve is conducted by (Churcher et. al, 1991). For our comparison, Berea sandstone sample-3 is used to compare with the simulation results since it has an absolute permeability, which matches with the one stated by (Dong & Blunt, 2009). The SNOW results are matching the experimental results within an acceptable range of accuracy. The Geodict simulation results are matching the experimental results in low mercury saturation, but they deviate as the mercury saturation increases.

Table 2: Parameters used to extract the Berea sandstone network and predict transport properties

Parameter	Value
SNOW R_max	10
SNOW Sigma	0.6
Pore entry pressure	0
Throat volume	0
η (Pore late filling)	0.85
S_w^*	0.214
P_c^* mode	Mean
Multiphase conductance mode	Medium
Trapping mechanism	Activated

Table 3: Fluid Properties used in Berea sandstone simulations

Parameter	Value
Mercury Interfacial Tension	480 mN/m
Mercury-Air Contact Angle	140 degrees
Water-Oil Interfacial Tension	30 mN/m
Water-Oil Contact Angle	8 degrees
Water Viscosity	1.05 cp
Oil Viscosity	1.39 cp

The results of relative permeability simulation, along with the comparison with experimental results, are shown in Figure 24. In this figure, the regions of the relative permeability curve which deviate from the experimental results are denoted by red circles. It can be seen that in high water saturations since the oil is not forming a connected phase; therefore, the relative permeabilities have large errors compared to the experimental results. This error is the same in case of water relative permeabilities for low water saturations. However, around the middle of the water saturation axis, we can observe a good match between the results of SNOW simulations and experimental results. These results can be used to match a Corey relative permeability model with a water Corey exponent of 4 and an oil Corey exponent of 2.5, which is demonstrated in Figure 25.

Geodict and SNOW simulation results of relative permeability are compared in Figure 26. In this comparison the trapping mechanism is turned off therefore, the relative permeability curve is different from Figure 24 and Figure 25. As can be seen in Figure 26, the capillary pressure of SNOW is higher, which is due to the pore late filling model implemented, and for relative permeability curves, SNOW is demonstrating a more accurate representation of oil relative permeability than Geodict. However, Geodict is more accurate in lower water saturation for water relative permeability.

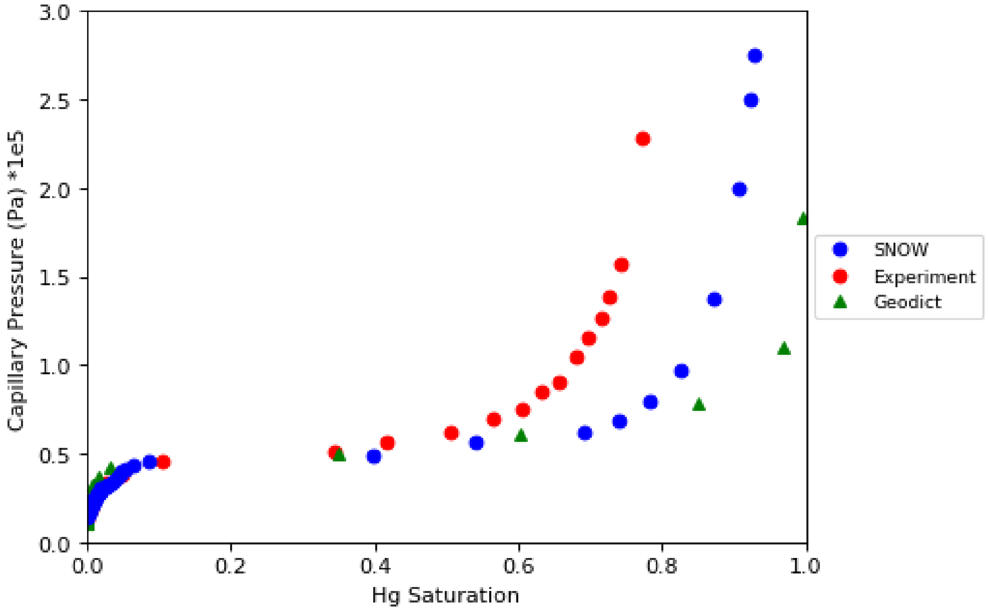


Figure 23: Capillary pressure data comparison for Berea sandstone

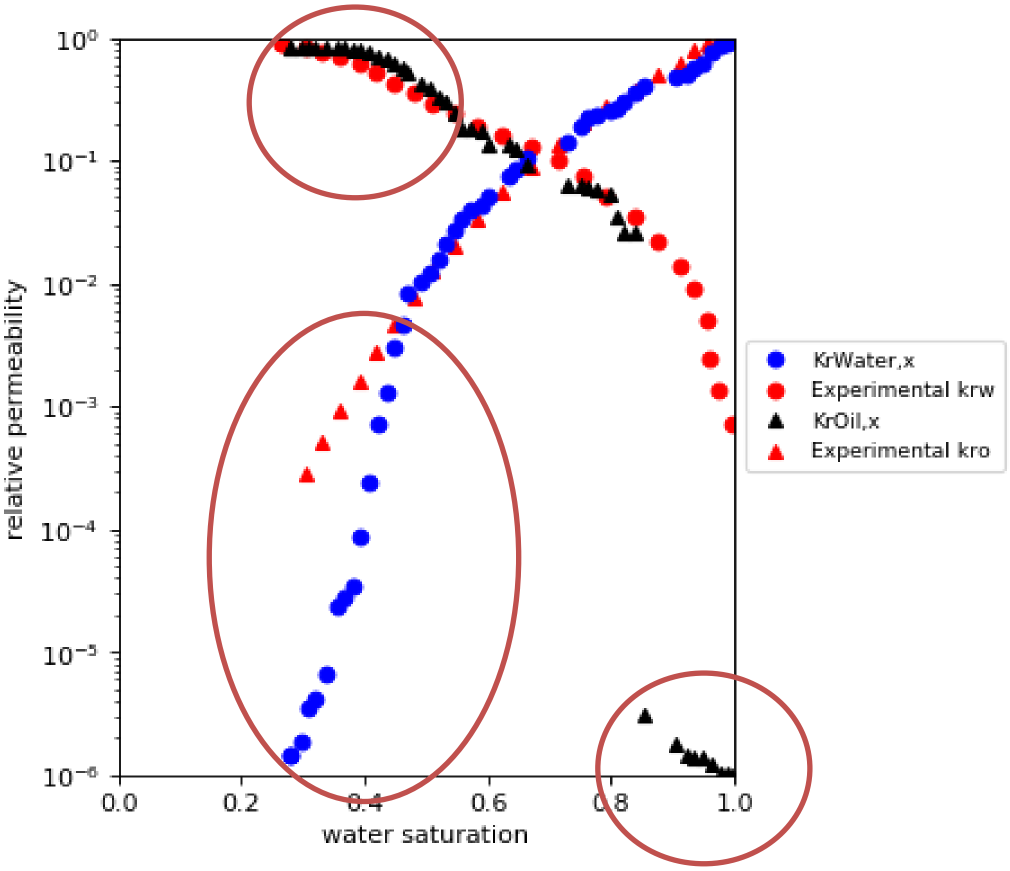


Figure 24: Comparison of relative permeability for SNOW simulation and experimental results

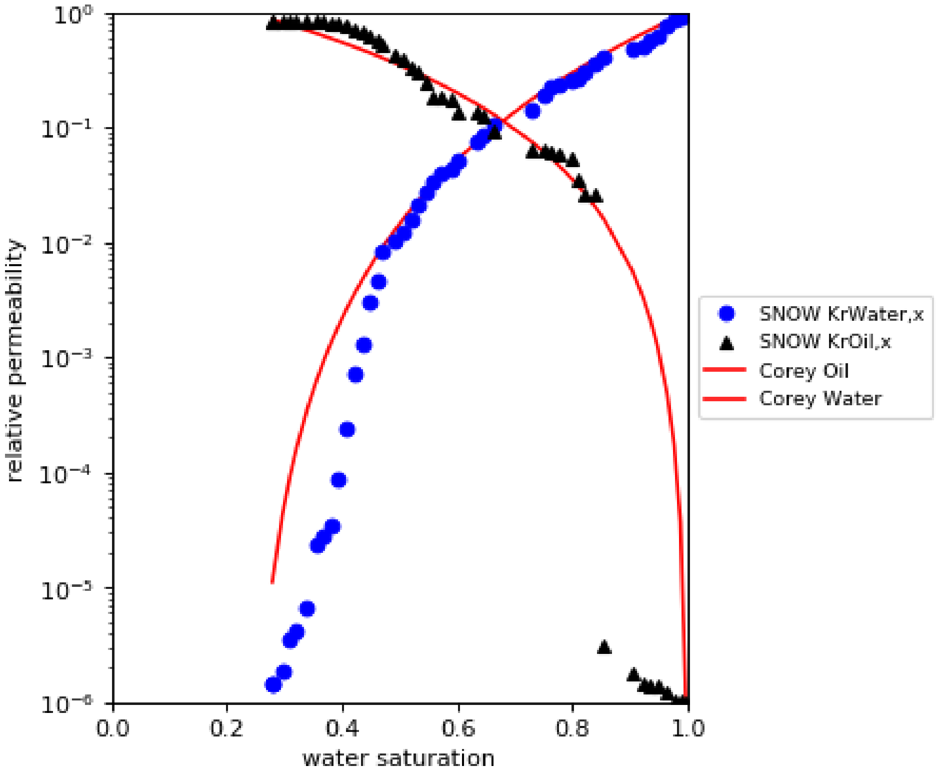


Figure 25: Corey match to relative permeability results of SNOW

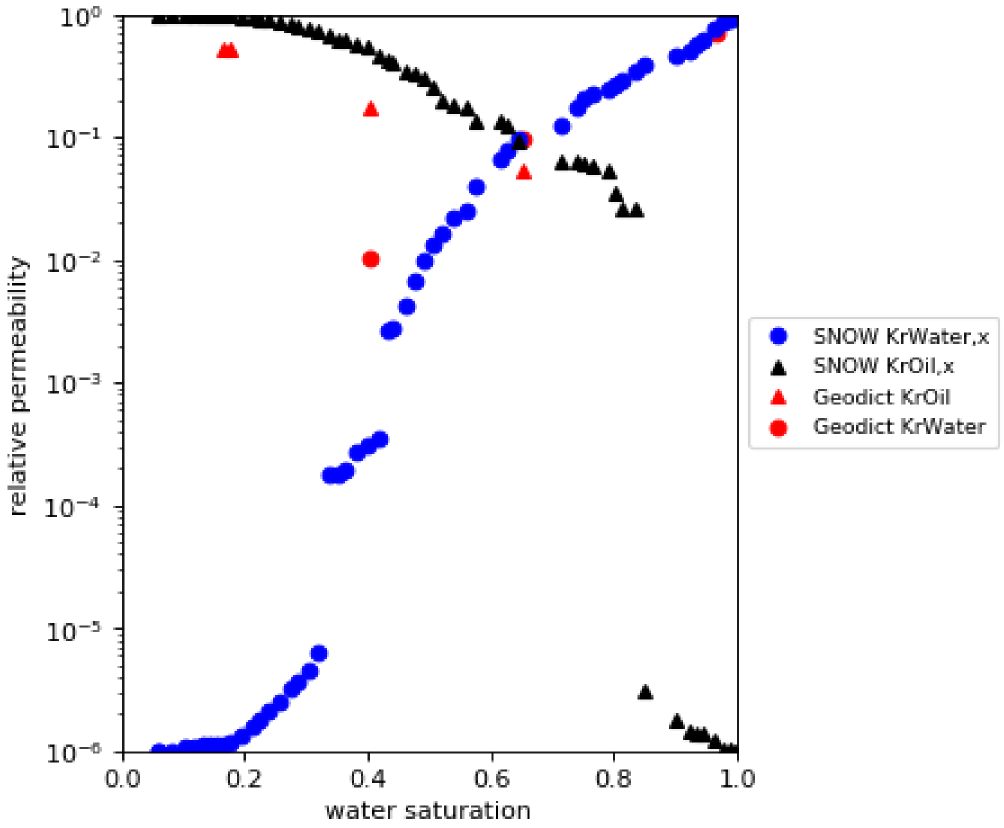


Figure 26: Comparison of SNOW and Geodict relative permeability results for Berea Sandstone

4.1.1 Sensitivity Analysis of Berea Sandstone

In this section, the sensitivity of different parameters of the SNOW simulation, which have an underlying uncertainty, is investigated in the simulation results. The reason for conducting this sensitivity analysis is to understand how accurate our simulation results are. Understanding the parameters that influence this uncertainty, we can improve the accuracy of the simulation results.

As can be seen in Figure 27 and Figure 28, the number of the pore and the number of the throats for Berea sandstone increase in the beginning and then decrease by increasing sigma and R max in the SNOW extraction algorithm. However, the results presented by J. Gostick (2017) state that the number of pore and throats should decrease when sigma and R max increase. The baseline for the number of the pores and the throats in the tornado charts are set to be the ones reported by J. Gostick (2017). In Figure 27 and Figure 28, the increase in the number of pores and the number of throats until R_max = 5 is interpreted to be artifacts. These artifacts can be a consequence of running the distance map in samples with resolution limitations. This means that for the case of Berea sandstone, a R_max > 5 should be chosen for further simulations.

In Figure 29, it can be seen that as reported by Bhattad (2011), the absolute permeability is less sensitive to the underlying structure, and the permeability variations with different sigma and R max parameters are a maximum of 10%. The mean variation in the absolute permeability is 3%, and the baseline for absolute permeability is set to be the one reported by J. Gostick (2017) for Berea sandstone.

The sensitivity of sigma and R max extraction parameters to the capillary pressure curve is shown in Figure 30. Although the extraction parameters are different, the capillary pressure curve is not very much different. In this figure experimental results are shown with red circles and different sets of R max and Sigma are shown with circles in random colors. Here it is important to mention that, comparing Figure 30 and Figure 31, we see different calculated points in the capillary pressure curve. This happens when we are having different R_max and Sigma extraction parameters, because each combination will result in different maximum points in the intrusion curve (Figure 17).

The sensitivity of the capillary pressure curve of Berea sandstone to the η parameter in the pore late filling model is shown in Figure 31 varying in the range of $0.75 < \eta < 0.95$. The results demonstrate that there is a small shift to the right for bigger η values. The capillary pressure curve for three different values of S_w^* is shown in Figure 32, in which $S^*{}_w$ is between 0.1 and 0.3. As the results show, for bigger S_w^* values, the curve will shift to the left. However, the sensitivity of the results to S_w^* is more than the sensitivity to η and SNOW extraction parameters.

The same set of sigma and R max, which was used in Figure 29, plus a variation of η parameter for late pore filling calculations in the range of 0.1, is used to derive the sensitivity of relative permeability curves, which are demonstrated in Figure 33 and Figure 34. It can be observed that the oil relative permeability has a more significant error compared to water relative permeability curves. The sensitivity of the relative permeability curves to the S_w^* parameter in the pore late filling model is demonstrated in Figure 35. It can be seen that bigger S_w^* values result in lower relative permeability values both for oil and water, and higher S_w^* values result in higher relative permeability values. The differences in relative permeability values are highlighted by a red circle in each relative permeability curve.

Figure 36 shows the sensitivity of different modes of P_c^* model on the capillary pressure and relative permeability results. Please note that the points calculated in the capillary pressure curve are different from the calculated points in the relative permeability curve. This is because we are using inscribed throat diameters to calculate capillary pressure curves points and equivalent throat diameters to calculate the relative permeability curve points. As can be seen in the case of using the min and mean mode, the capillary pressure is underestimated, and in the case of the max mode, the capillary pressure is fitting the experimental results. As for the relative permeability, there will be an underestimation of oil relative permeability in case of using the min mode. However, there will be an overestimation of oil relative permeability for the max mode. Nevertheless, since capillary pressure curves and relative permeability curves are connected, we have to be consistent in choosing a general mode to adapt our capillary pressure. This means that the same mode of capillary pressure adaptation has to be selected for both capillary pressure simulations and relative permeability simulations. For the Berea sandstone is was decided to use the mean mode as it can predict both capillary pressure and relative permeability within an acceptable error.

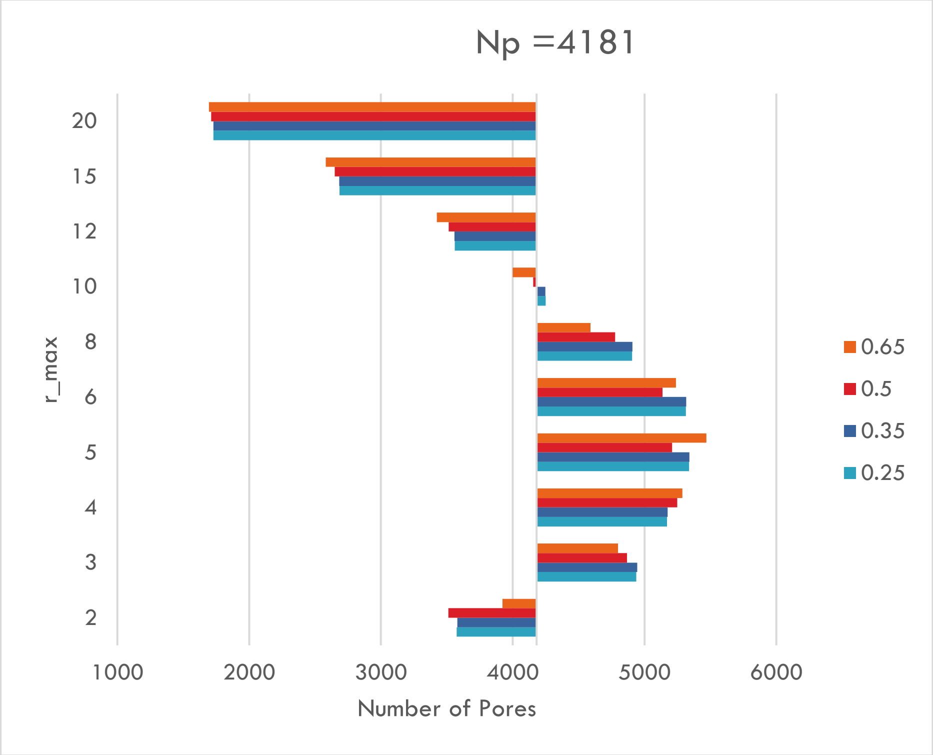


Figure 27: Tornado chart representing the sensitivity of the number of the network pores based on sigma and R max parameters of SNOW for Berea sandstone

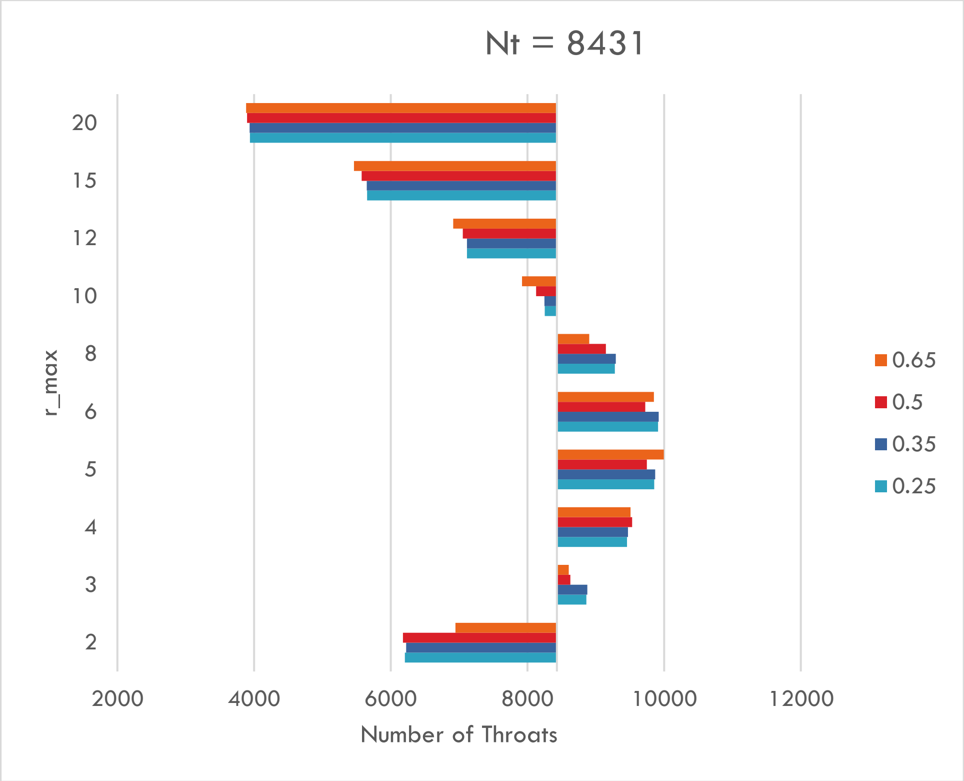


Figure 28: Tornado chart representing the sensitivity of the number of the network throats based on sigma and R max parameters of SNOW for Berea sandstone

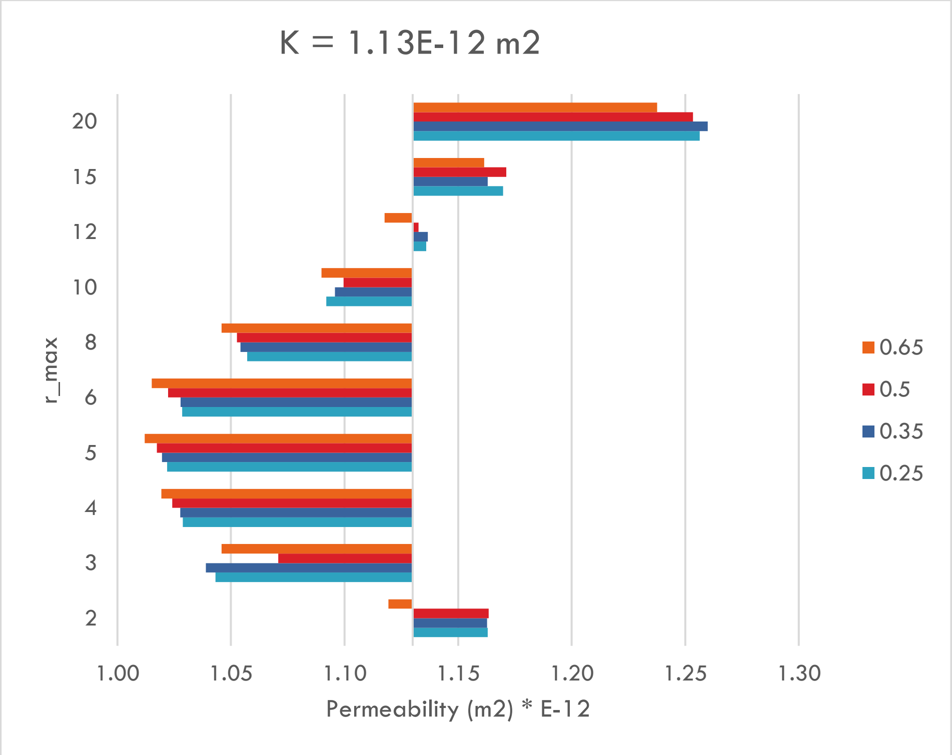


Figure 29: Tornado chart representing the sensitivity of the absolute permeability based on sigma and R max parameters of SNOW for Berea sandstone

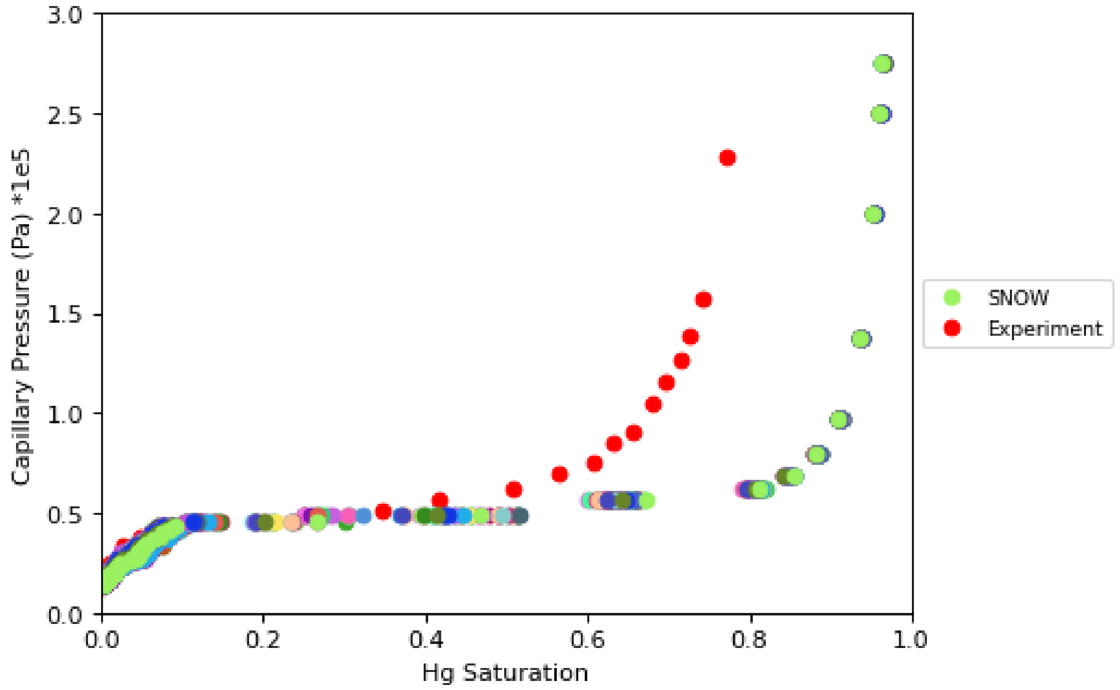


Figure 30: Sensitivity of the capillary pressure curve on sigma and R max for SNOW for Berea sandstone (points are lying on each other)

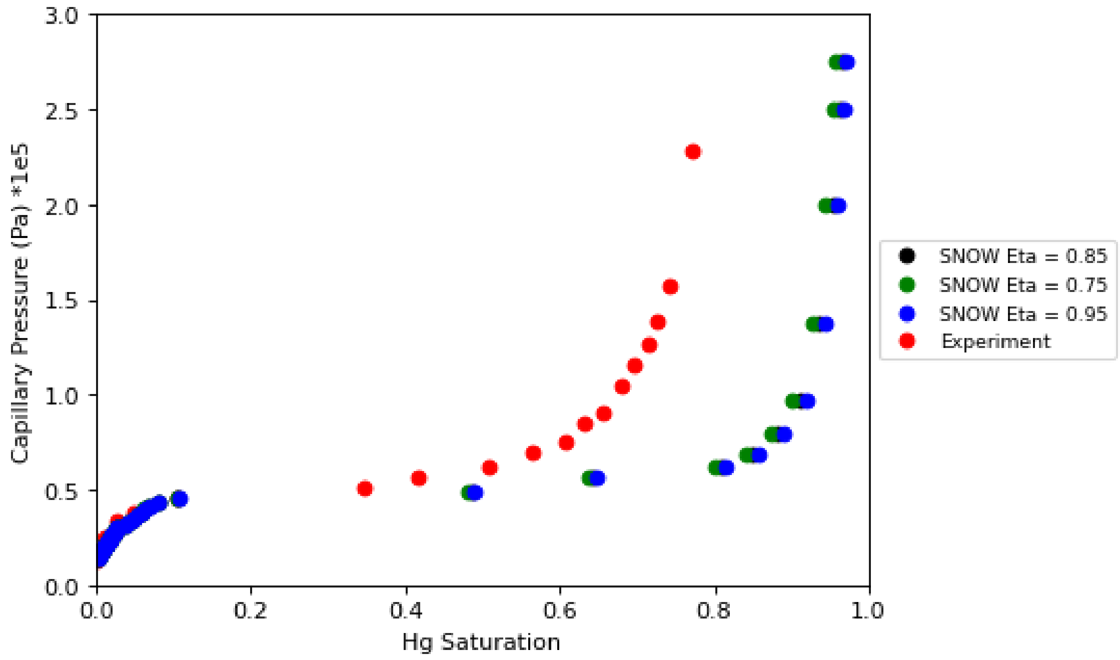


Figure 31: Sensitivity of Berea sandstone capillary pressure to η pore late filling parameter

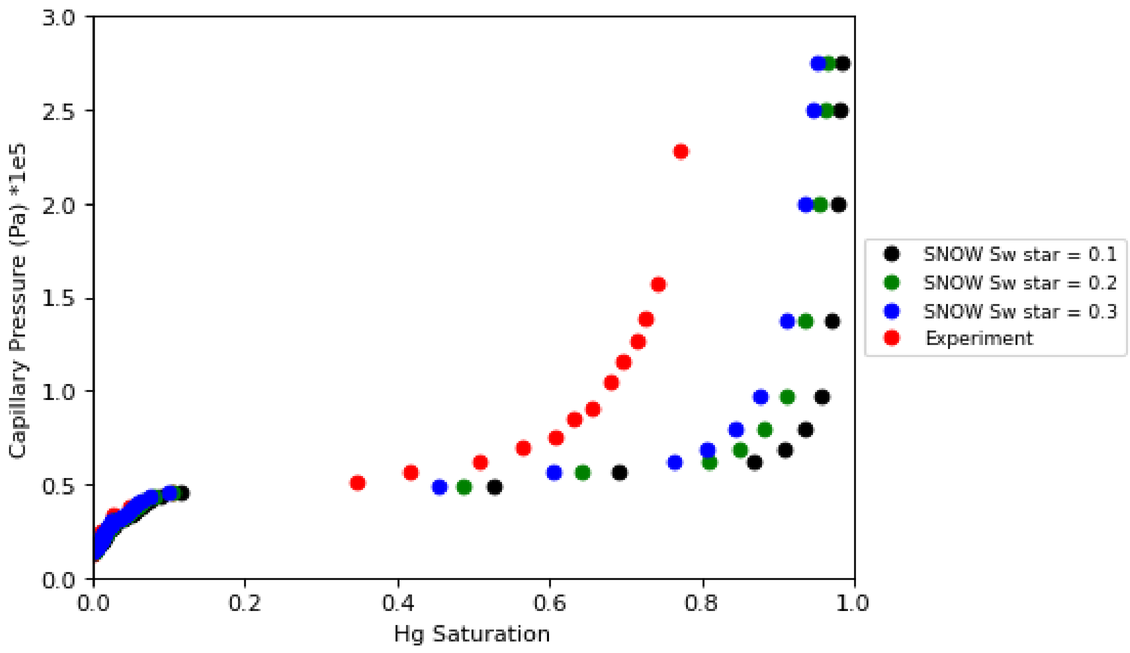


Figure 32: Sensitivity of capillary pressure curve of Berea sandstone on S_w^* parameter of pore late filling model

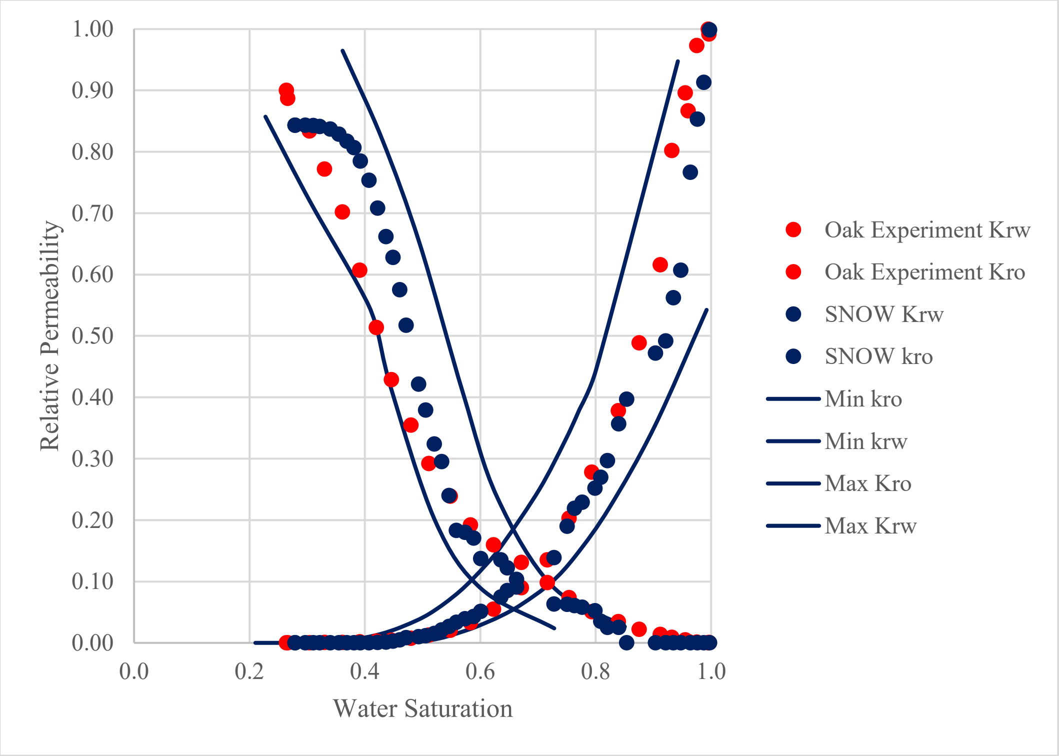


Figure 33: Sensitivity of relative permeability curve on sigma and r max and η parameter in pore late filling for Berea sandstone (linear scale)

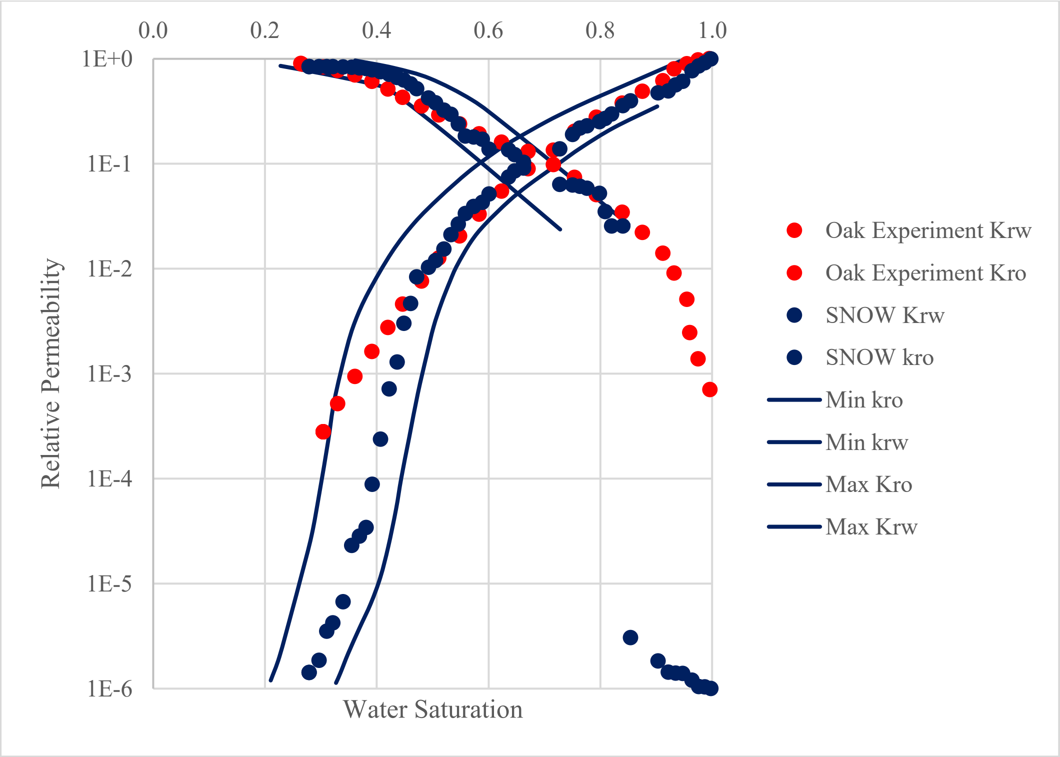
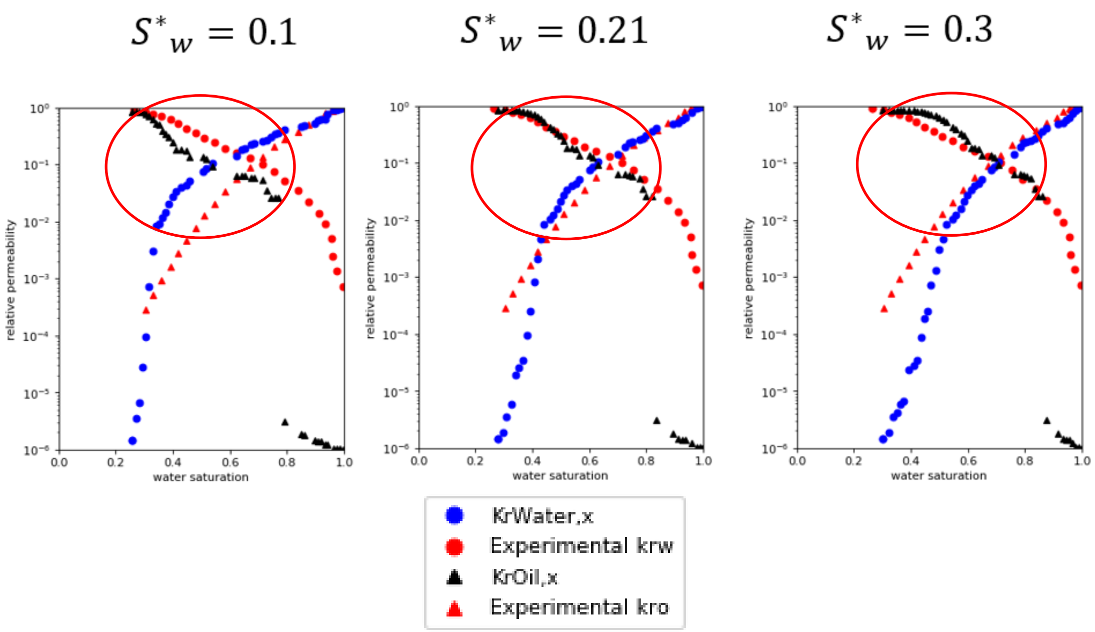


Figure 34: Sensitivity of relative permeability curve on sigma and r max and η parameter in pore late filling for Berea sandstone (logarithmic scale)



$S^*{}_w = 0.1$ $S^*{}_w = 0.21$ $S^*{}_w = 0.3$

Figure 35: Sensitivity of relative permeability curves of Berea sandstone on S_w^* parameter of pore late filling model

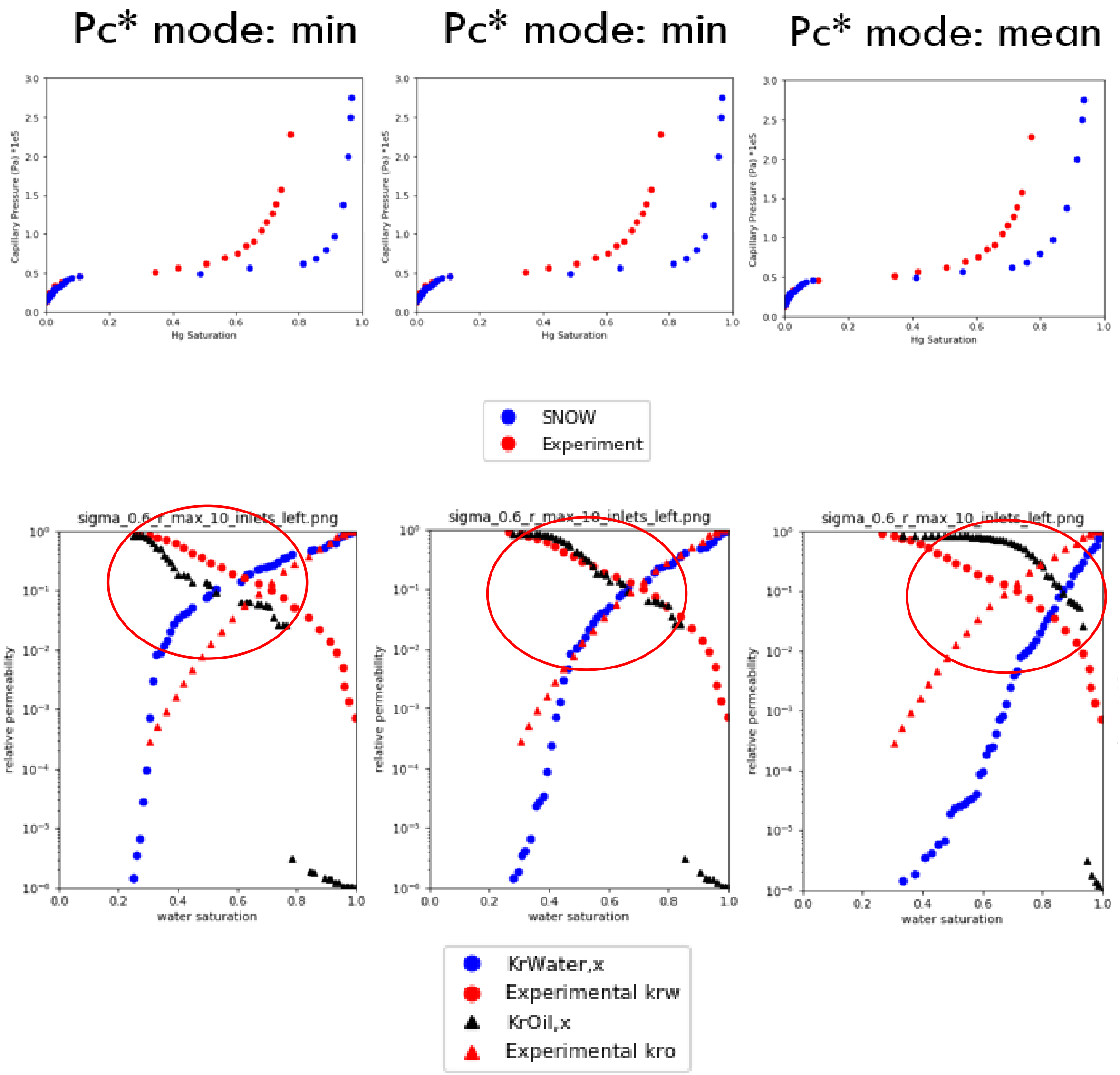


Figure 36: Sensitivity of capillary pressure curves and relative permeability curves for different P_c^* mode in the pore late filling model of Berea sandstone, please note the different points in the relative permeability and capillary pressure curves due to different throat diameter implementation

Until here, the simulations were conducted by making the assumption that the pore volumes are the summations of the volumes of voxels, which are included in a region corresponding to a pore. These regions are defined after the segmentation of the pore space using the SNOW algorithm. Therefore, pore late filling parameters were derived and implemented using the available fractal information of the rocks from the literature and the initial fraction from the MICP experimental data.

Another method to calculate the pore volumes is to calculate the volume of spheres using the inscribed pore diameters calculated by the SNOW algorithm. In the case of using this method to calculate the pore volumes, a lot of porosity of the pore space will be lost. However, it was observed that still good results could be obtained from the simulator if we use the proper parameters for the pore late filling model. The η, in this case, is one since we are dealing with three-dimensional spherical pore volumes, and the initial fraction is the amount of pore space lost during the calculations of the pore volumes in the last step. The advantage of this method is that when we want to simulate the rocks from which we do not have a lot of data available, we have an easier choice for pore late filling model parameters. This advantage is explained further in detail.

In the case of Berea sandstone, only 9 % of the porosity is captured by calculating the summation of the volumes of the pores from the pore diameters. Therefore, we can use the initial fraction (S_w^*) of 0.91 in the pore late filling formula, which is the volume of the pore space not captured by the spherical pore geometry compared to the porosity obtained from voxel count method. Also, here η is one since we are assuming spherical shape for our pores. During this study, it was found that using this method of the simulation, the P_c^* mode of 'min' works best for all of the simulation samples.

Figure 37 and Figure 38 demonstrate the capillary pressure and relative permeabilities derived from this simulation method for Berea sandstone. It can be seen that the capillary pressure curve has a closer match relative to the capillary pressure derived in Figure 23. Also, the relative permeabilities show results which are matching the experimental results as good as the ones reported in Figure 24.

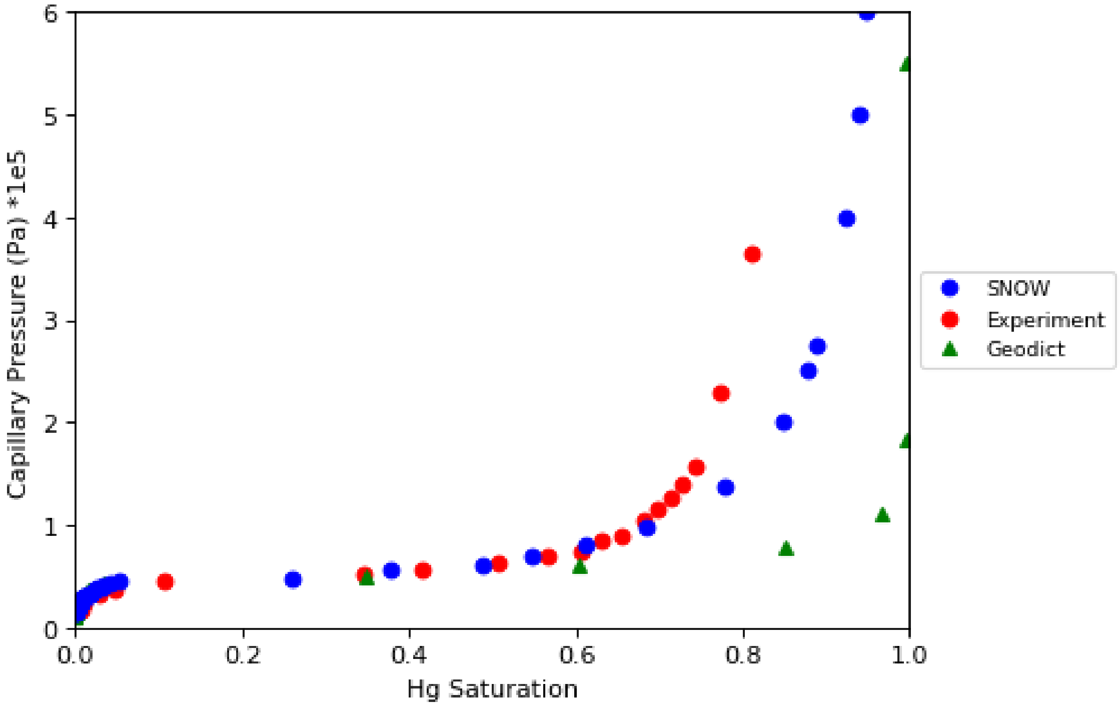


Figure 37: Capillary pressure of Berea sandstone using pore diameters to derive pore volumes

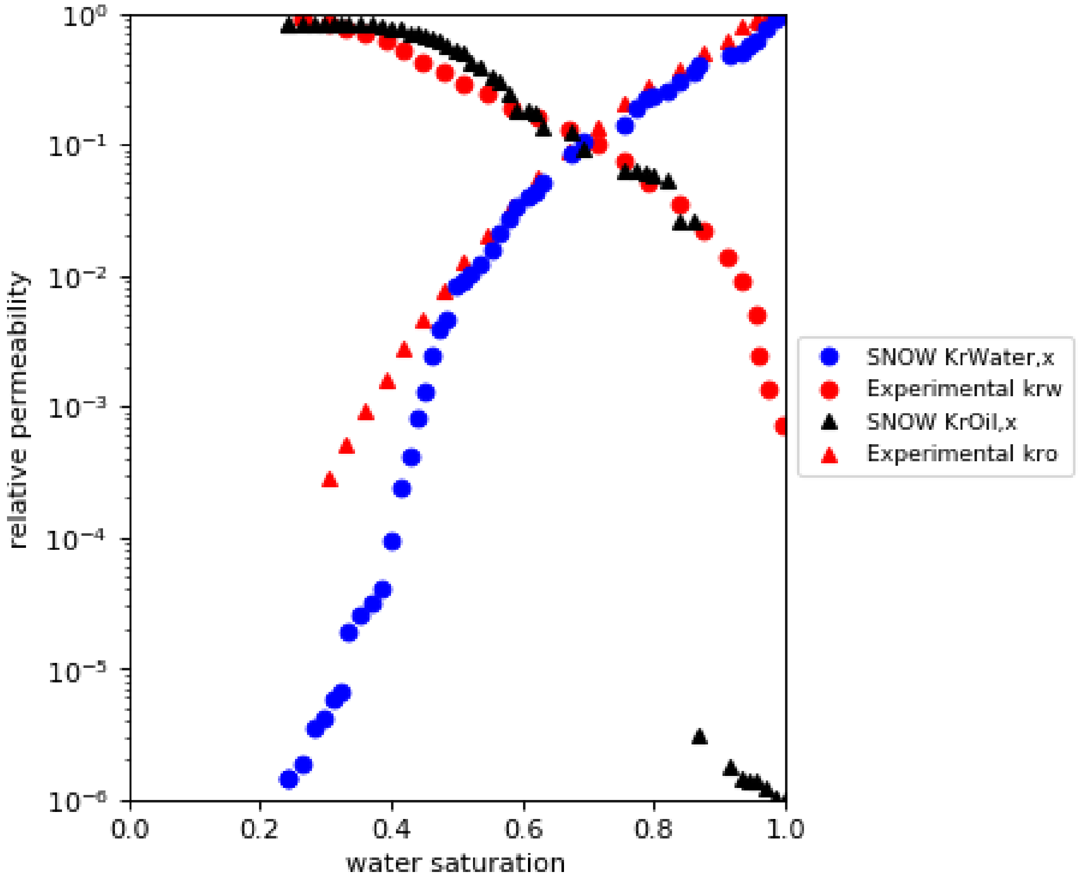


Figure 38: Relative permeability of Berea sandstone using pore diameters to derive pore volumes

4.2 Bentheimer Sandstone

As another sample to validate our model with the experimental results, Bentheimer sandstone is used. Bentheimer is a well-sorted and water-wet sandstone which is mainly made of quartz, feldspar, and authigenic clays. The Amott wettability index of the experimental results is between 0.08 and 0.1, which accounts for a water-wet state (Oren et al., 1998). The micro CT image of Bentheimer has a resolution of 3.0035 μm and a porosity of 0.21 and a size of 1000^3 voxels. In this work, due to computational limitations, a crop of 500^3 voxels was used, starting from the zero coordinate point of the main image (Muljadi, n.d.; Muljadi et al., 2015). Figure 39 demonstrates a 2D cross-section of the Bentheimer micro CT image, and Figure 40 illustrates the pore network extracted using the SNOW algorithm from a crop of the main micro-CT image.

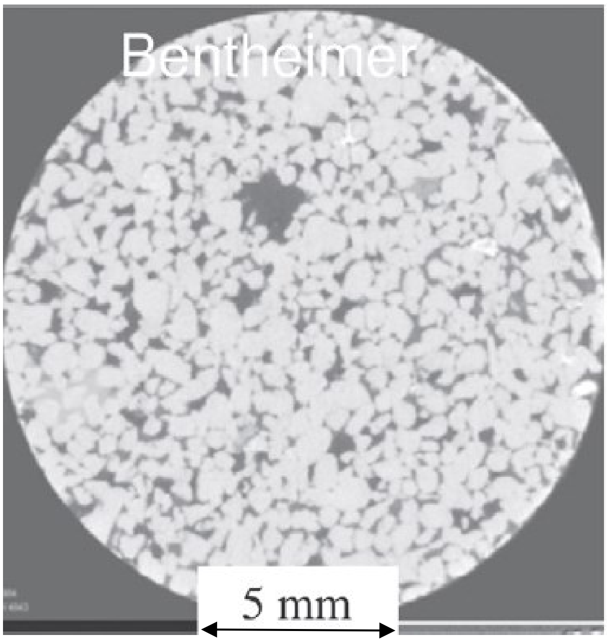


Figure 39: 2D cross-section of Bentheimer micro CT image (Muljadi et al., 2015)

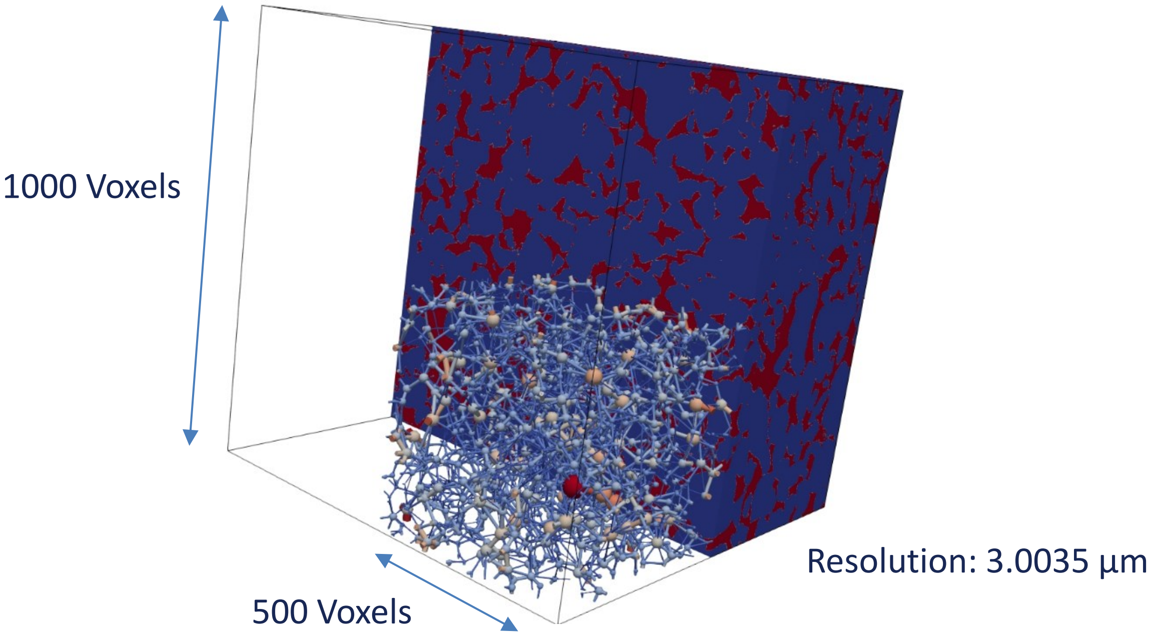


Figure 40:Pore network extracted from Bentheimer sandstone using SNOW algorithm

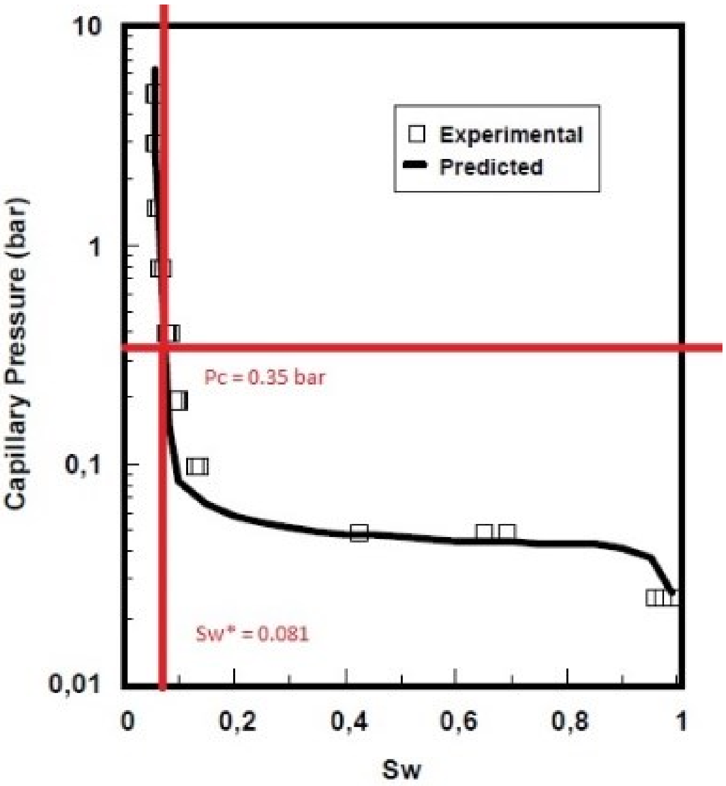


Figure 41: Obtaining S_w^* to implement in Bentheimer sandstone modeling (Oren et al., 1998)

Table 4: Pore networking properties of Bentheimer sandstone

Parameter	Value
Permeability from SNOW	2002 mD
Permeability from Geodict	3516 mD
Experimental Permeability	2840 mD
Resolution	3.0035 µm
Porosity	0.21
SNOW residual saturation	0.213
Average Coordination Number	3.56

As it can be seen in Table 4, SNOW predictions for absolute permeability are underestimating the experimental permeability reported by Øren et al. (1998) and the Geodict results are overestimating the absolute permeability. However, they are both in the same order of magnitude. Nevertheless, in the case of a rock sample with an absolute permeability of several Darcies, the same differences in measurements also appear in experimental results from the laboratory.

In the case of Bentheimer sandstone, P_c^* mode is set to min and the consequence of using this method is that, there is a better match with the experimental results and SNOW predictions. These parameters and SNOW network extraction parameters are demonstrated in Table 5. It is essential to mention that S_w^* is chosen from Figure 41. The calculations which are denoted on this figure are obtained from equation (4.2). In this figure, Pc = 0.35 represents the maximum observable capillary pressure due to the resolution limit of the image, which is 3.0035 μm. The η for Bentheimer sandstone is chosen based on the findings of Zhang et al. (2018), who calculated the fractal dimensions of Bentheimer sandstone based on the mercury injection experiment.

$$0.35 * 1E5 = -\frac{2 * 0.035\, cos\,(140)}{(\mathbf{3.0035} * 1E - 6)/2} \tag{4.2}$$

Using the fluid properties, which are shown in Table 6, capillary pressure and relative permeability predictions of SNOW are compared with experimental results in Figure 42 and Figure 43. In the predictions of capillary pressure, the SNOW is generally underestimating the experimental capillary pressure, which can be due to the differences in the contact angles of the real rock sample. Also, in higher oil saturations, the results deviate from experimental results, which can be due to the inaccuracy in residual saturation predictions of the model from reality and errors rising from pore late filling model.

As for the relative permeability curve shown in Figure 43, the relative permeabilities for oil are erroneous in low oil saturations since the oil has not formed a connected phase. For water relative permeability, the same phenomena are happening at low water saturations. Generally, the model has a better ability to predict the oil relative saturation at higher oil saturations and water relative permeability at higher water saturations. This behavior of the simulation results is also the same as observed in the Berea sandstone.

The comparison of results between SNOW transport predictions and Geodict predictions is shown in Figure 44 and Figure 45. It can be seen that in both curves, the results are very close. However, for the higher mercury saturation in the capillary pressure curve, there is a mismatch between the results which is due to the implementation of pore late filling model in SNOW modeling.

Table 5: SNOW network extraction parameters for Bentheimer sandstone

Parameter	Value
SNOW R_max	12
SNOW Sigma	0.6
Pore entry pressure	0
Throat volume	0
η (Pore late filling)	0.67
S_w^*	0.081
P_c^* mode	Min
Multiphase conductance mode	Medium
Trapping mechanism	Deactivated

Table 6: Fluid properties used to simulate the flow in Bentheimer sandstone

Parameter	Value
Mercury Interfacial Tension	480 mN/m
Mercury-Air Contact Angle	140 degrees
Water-Oil Interfacial Tension	35 mN/m
Water-Oil Contact Angle	40 degrees
Water Viscosity	1.06 cp
Oil Viscosity	1.4 cp

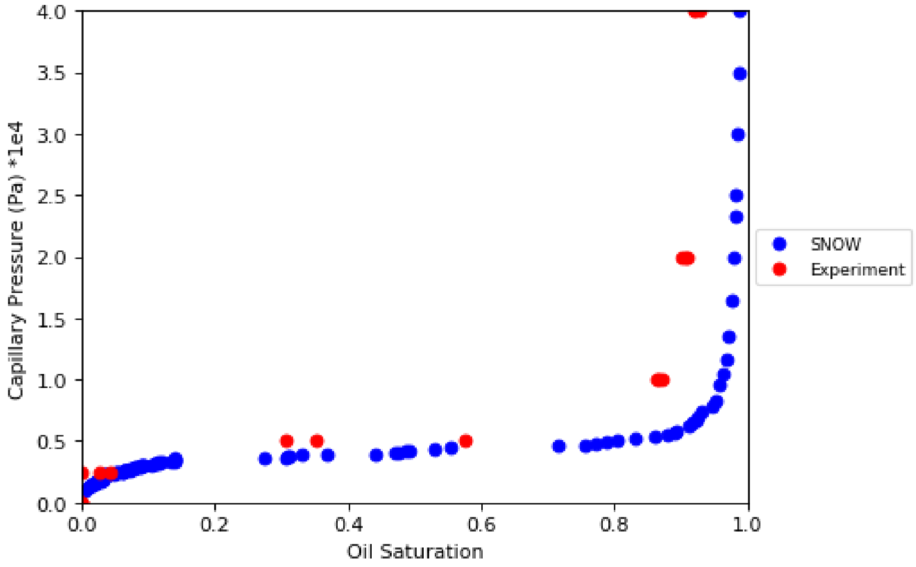


Figure 42: Capillary pressure comparison of SNOW predictions and experimental results of (Oren et al., 1998) for Bentheimer sandstone

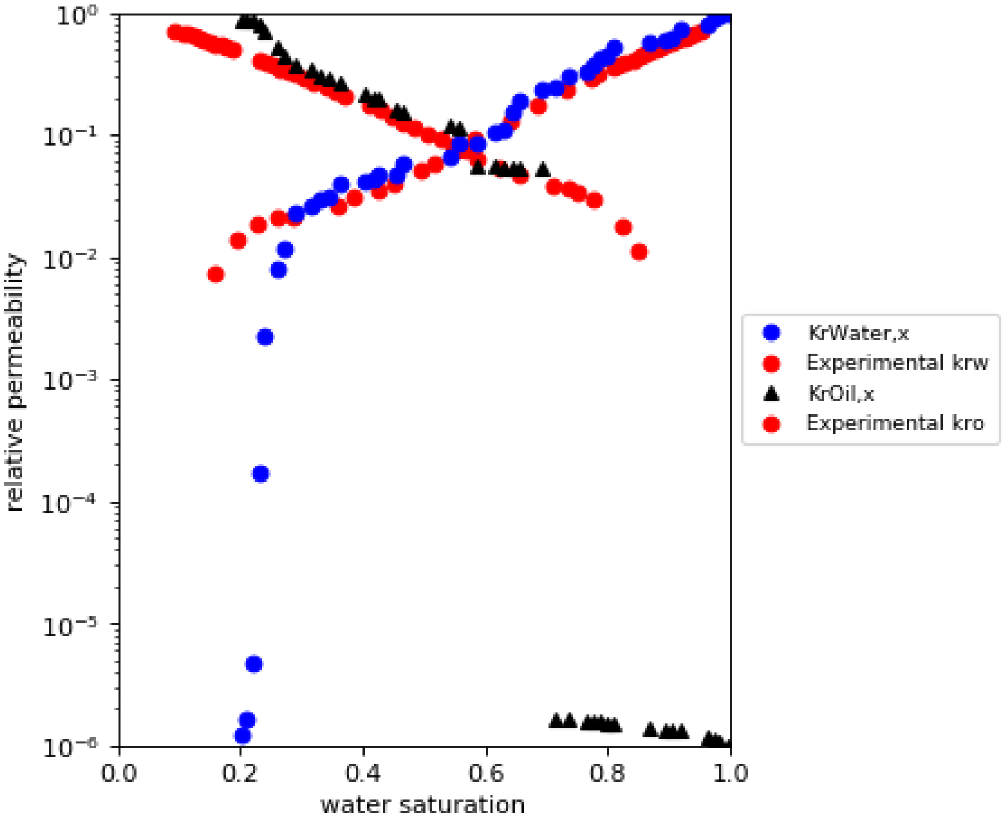


Figure 43: Relative permeability comparison of SNOW predictions and experimental results from (Oren et al., 1998) for Bentheimer sandstone

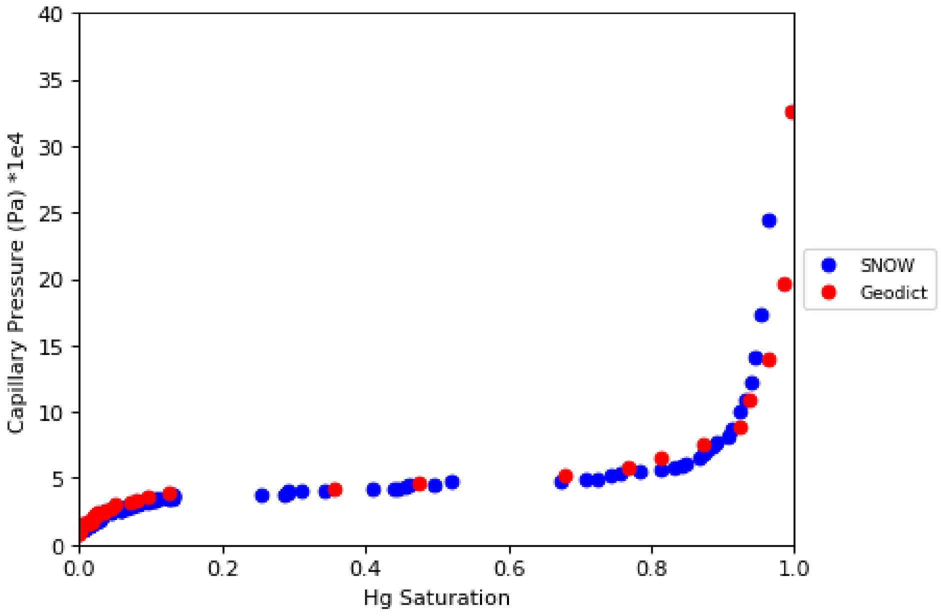


Figure 44: Capillary pressure comparison of SNOW predictions and Geodict results for Bentheimer sandstone

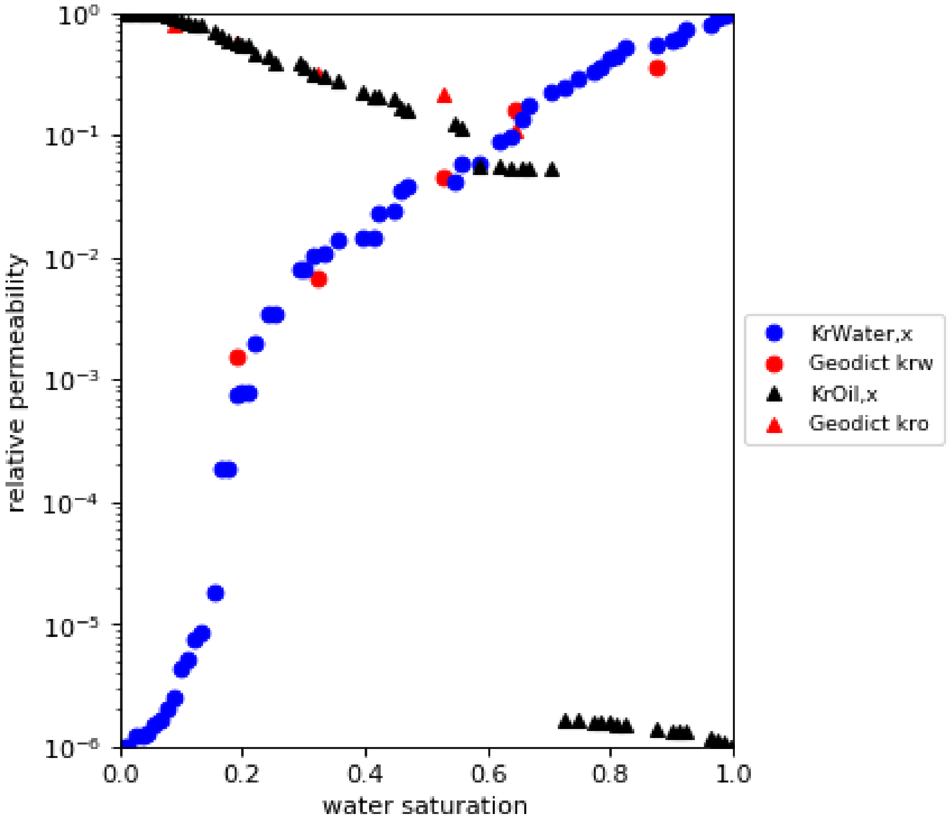


Figure 45: Relative permeability comparison between SNOW predictions and Geodict results for Bentheimer sandstone

4.3 Interfacial Tension and Contact Angle

In a rock sample, the contact angle and interfacial tension between the fluid phases are not just a constant value. In this part, the sensitivities of the primary drainage relative permeability with respect to interfacial tension and contact angle are investigated.

In order to analyze the effect of changing the interfacial tension and the contact angle on the relative permeability curves, the underlying algorithm needs to be changed. This is because the primary drainage process in OpenPNM is modeled with the percolation theory, and in the framework of OpenPNM, we are dealing with invasion sequences at each percolation step. Later we use this invasion sequence to derive our relative permeability curves, and these curves only change if the invasion sequence is different. The invasion sequence is observed to be solely dependent on the structure of the pore network itself. Therefore, it changes if we use different extraction parameters to get a different pore network.

Based on our observations, changing the values of the Young-Laplace equation in OpenPNM does not change the invasion sequence. Therefore, the capillary pressure curve is still having the same saturation points but with different capillary pressures. The difference in the capillary pressure is then dependent on the values of interfacial tension and the contact angle that were used in the Young-Laplace equation. This difference is depicted in Figure 46.

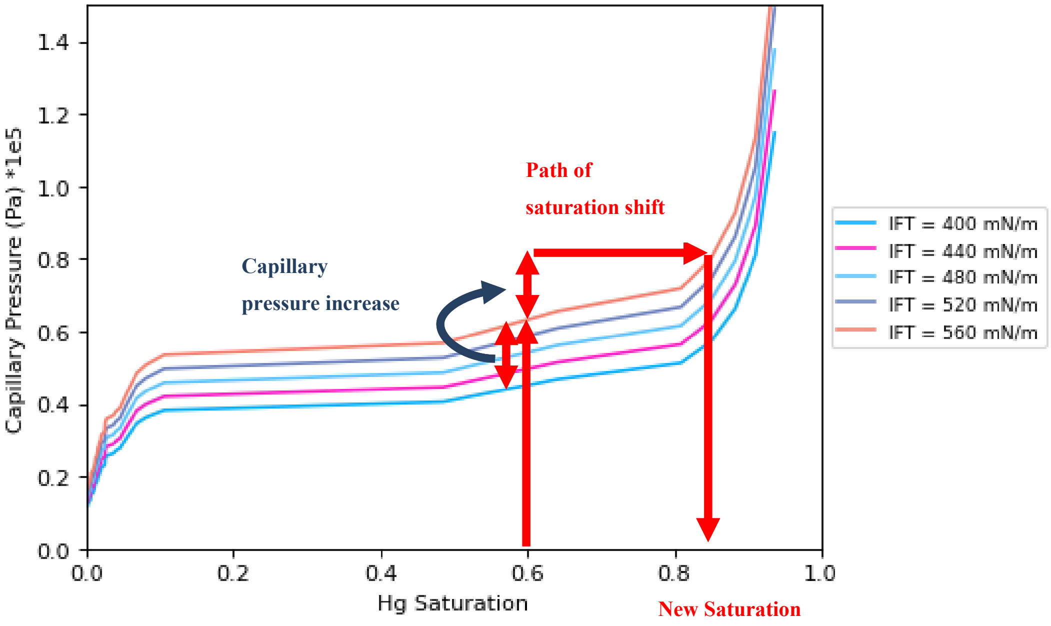


Figure 46: Capillary pressure of Berea sandstone for mercury injection at different interfacial tensions

In the new algorithm of calculating the relative permeabilities, the saturation which is corresponding to an invasion state and a certain capillary pressure is shifted based on the shapes of the capillary pressure curves. For example, In Figure 46 it can be seen that in the case we change the IFT from 560 mN/m to 400 mN/m, the capillary pressure curve will move down based on the Young-Laplace equation. Therefore, if we have already calculated the relative permeability curve for an IFT of 560 mN/m, which represents in this example our base case, then by changing the IFT to 400 mN/m, we shift the saturation based on the red arrows indicated in Figure 46. A limited saturation shift physically makes sense because in reality, in the case of having less IFT then less snap offs occur and smaller pores are invaded more easily, resulting in a higher invading phase saturation.

In terms of the implementation, in the loop of calculating the relative permeabilities, after choosing the new parameters to calculate the Young-Laplace equation, the saturation of the new calculated capillary pressure is recalculated based on the capillary pressure curve of the base case. The base case for this sensitivity analysis is the case with which the simulation results were matching the experimental data.

In this investigation, two Berea sandstone models from the same rock type, a Bentheimer sandstone and sandstone one, were investigated. In this analysis, the relative permeabilities under different conditions were matched with the extended Corey model for relative permeabilities. The match between the relative permeability curves and the Corey exponent for Berea sandstone is shown in the Appendix B. The code to simulate this sensitivity is shown in the Appendix A. The match between the relative permeability curves and the Corey exponents was done visually and based on our subjective decision and weighting on which simulation results are more accurate. The trapping mechanism for this simulation has been turned off to observe the match more clearly.

Figure 47 and Figure 48 demonstrate the variations of the oil and water exponent in Corey extended relative permeability model for different rock samples versus different contact angles. The trend shows an increase in the oil exponent and a decrease in the water exponent with the increase in the contact angle. The changes of residual water saturations for the range of the investigated contact angles are shown in Figure 49. It can be seen that, as the contact angle increases the residual saturation decreases, which is a result of the model going towards a neutral wetting state. This is a correct trend since with the increase of the contact angle, the mobility of the wetting phase should be increasing and the mobility of the non-wetting phase should be decreasing.

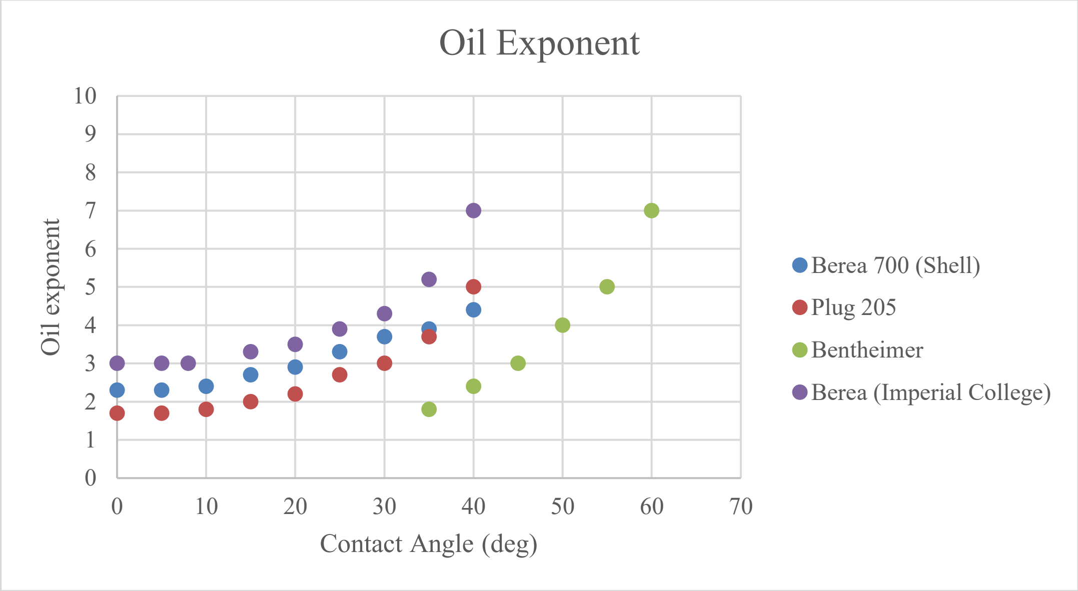


Figure 47: Variation of the oil exponent in Corey extended relative permeability model for different rock samples versus different contact angles

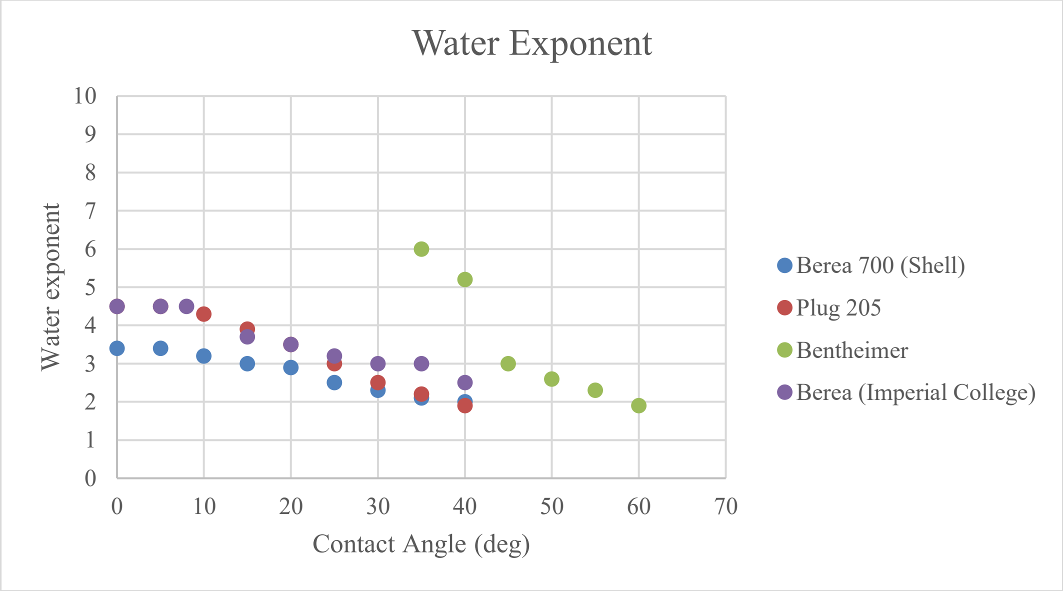


Figure 48: Variation of the water exponent in Corey extended relative permeability model for different rock samples versus different contact angles

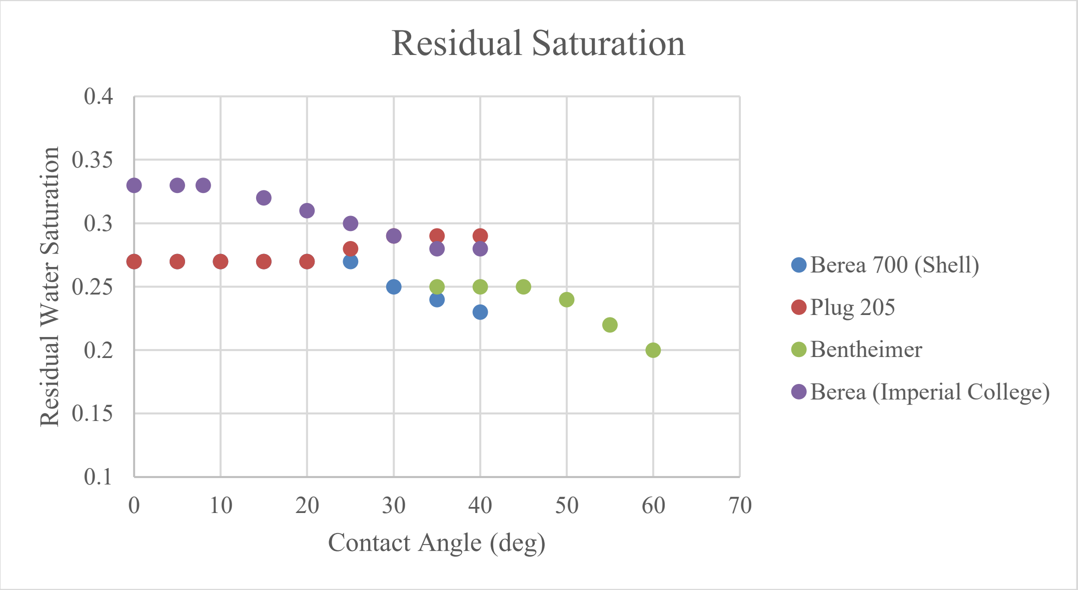


Figure 49: Variation of the residual water saturation in Corey extended relative permeability model for different rock samples versus different contact angles

Figure 50 and Figure 51 demonstrate the variation of the oil and water exponents in Corey's extended relative permeability model for different rock samples versus different interfacial tensions. The trends show a decrease in the oil exponent and an increase in the water exponent with the increase of the interfacial tension.

Figure 52 shows the sensitivity of the residual water saturation to the interfacial tension. It can be seen that as the interfacial tension increases, the residual water saturation also increases. The reason that the increase in the residual water saturation here is more evident compared to the variations of the contact angle is the Young-Laplace equation. In this equation, capillary pressure is having a linear relationship with the interfacial tension whereas, the relationship with the contact angle is given in the form of a cosine function.

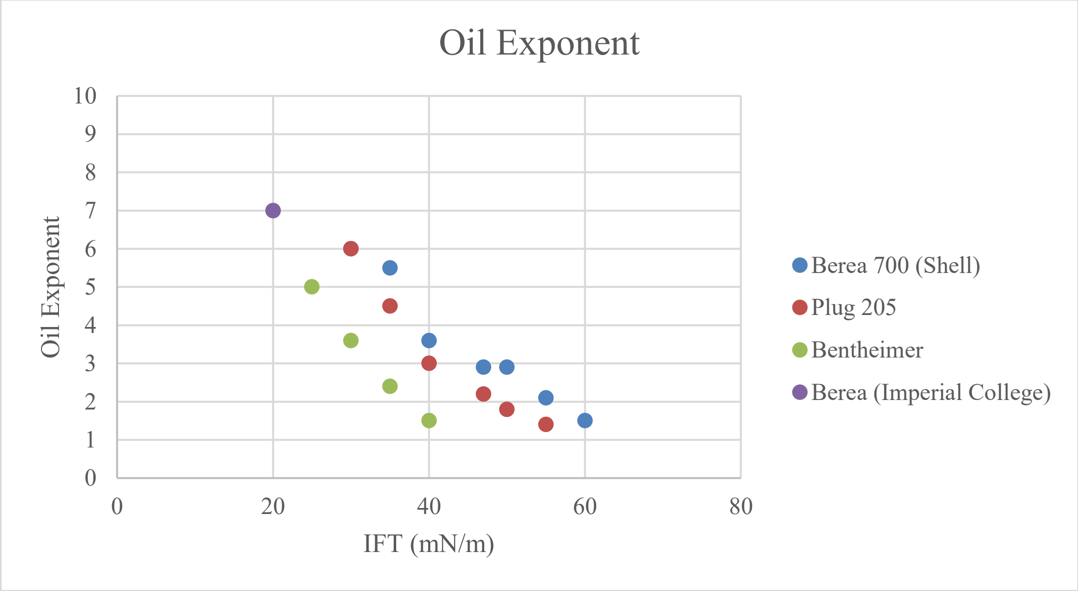


Figure 50: Variation of the oil exponent in Corey extended relative permeability model for different rock samples versus different interfacial tensions

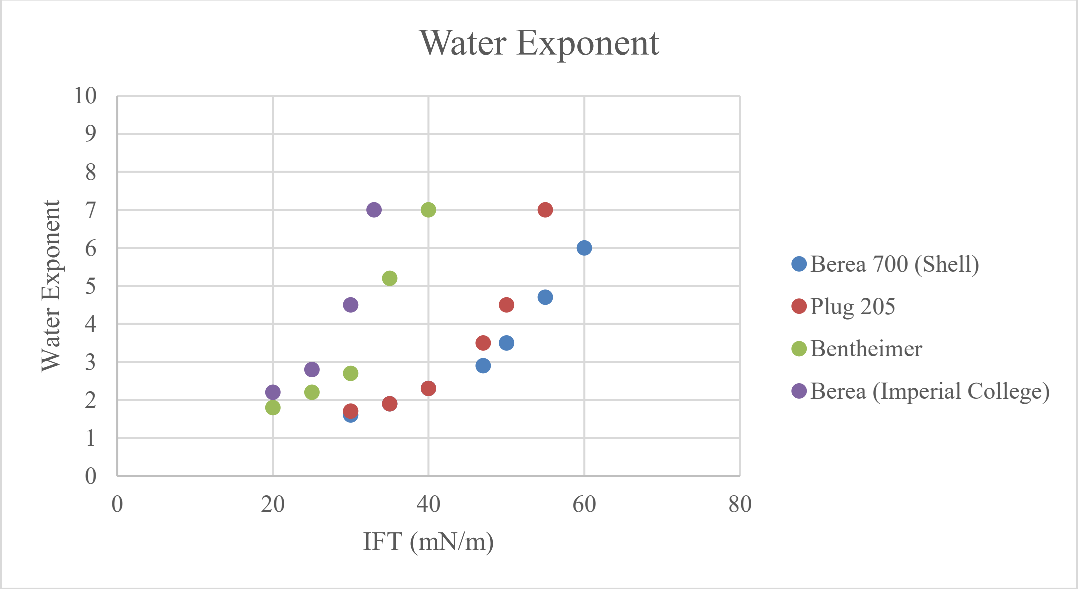


Figure 51: Variation of the water exponent in Corey extended relative permeability model for different rock samples versus different interfacial tensions

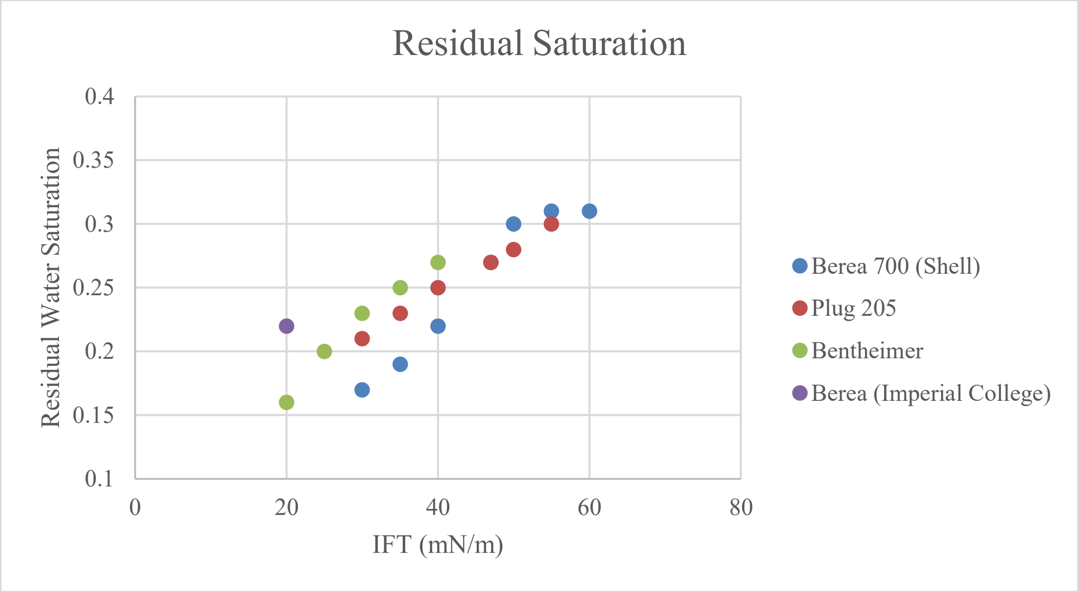


Figure 52: Variation of the residual water saturation in Corey extended relative permeability model for different rock samples versus different interfacial tensions

Chapter 5

Results and Discussion

5.1 Application

After validating our model using the validation samples, the model is used to predict the properties of a sandstone hereby and after called sandstone 1. The original image size is 650 * 650 * 1200 voxels, and the voxel size is 3.82 µm. However, due to computational limitations, a cube with the size of 500 voxels was cropped from the main image, starting from the zero coordinate points. The voxel count porosity of the cropped image is 0.27. A cross-section of the sandstone 1 segmented micro-CT image is shown in Figure 53 in which the black color demonstrates the matrix, and white shows the pore space. The pore network extracted from the sandstone 1 micro-CT image is shown in Figure 54.

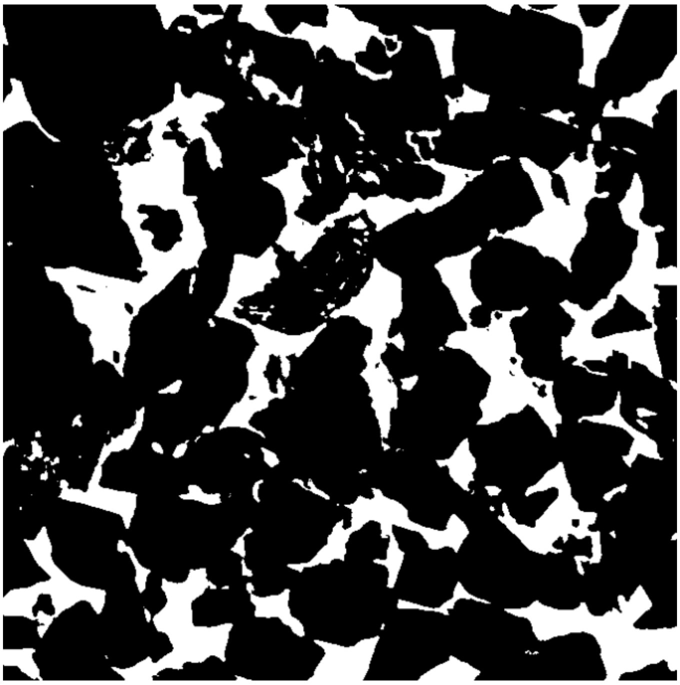


Figure 53: The segmented micro-CT image of sandstone 1

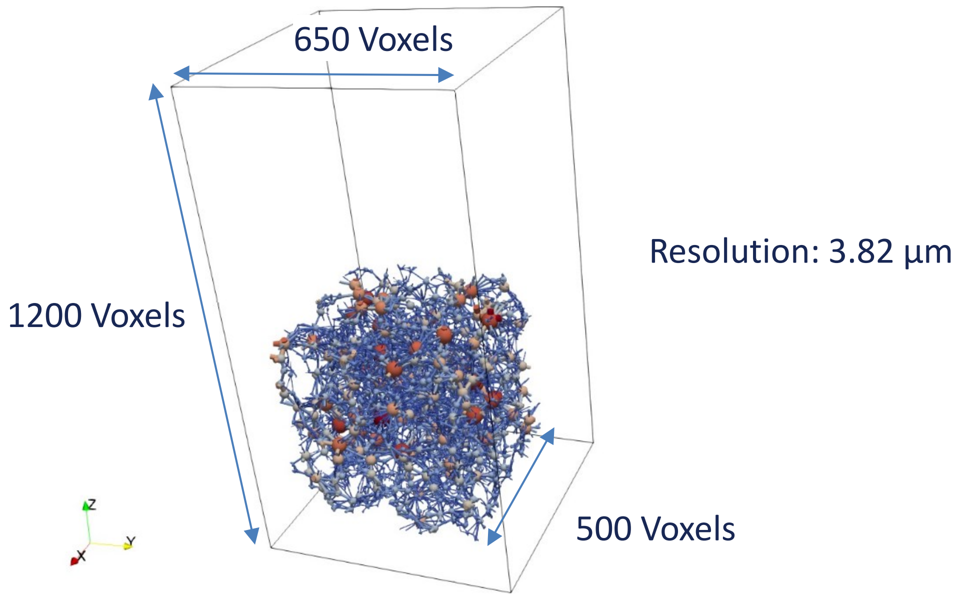


Figure 54: Pore network extracted from sandstone 1

Since the fractal dimension of sandstone 1 is undefined, we use our alternative pore volume, which is the pore volume derived from the inscribed pore diameter. In this calculation, the pore volumes are the volume of a sphere with inscribed pore diameters, whereas, in case of having more data, the pore volumes are calculated by summing the volumes of the voxels which fill the pore region after watershed segmentation by SNOW. Calculating the pore volumes of spheres in the pore space, pore volumes will be 16% of the total voxel count porosity. Therefore, we will have an initial fraction of 0.84 (1 – 0.16) and η = 1 for the pore late filling model. Moreover, as recommended before for this method of pore late filling model implementation, P_c^* mode of 'min' is used for the simulations.

Using the network and fluid properties demonstrated in Table 8 and Table 9, the transport prediction of our modeling for sandstone 1 is shown in Table 9. It can be seen that absolute permeability is underestimated by approximately a factor of two in our modeling compared to the reported experimental results. However, in this case, Geodict is overestimating the absolute permeability but with better accuracy.

Table 7: Transport prediction results for sandstone 1

Parameter	**Value**
Permeability from SNOW	6360 mD
Permeability from Geodict	~12000 mD
Experimental Permeability	10995 mD
Resolution	3.82 μm
Porosity	0.27
SNOW residual saturation from trapping mechanism	0.19

Table 8: Network parameters implemented for sandstone 1 simulations

Parameter	**Value**
SNOW R_max	5
SNOW Sigma	0.5
Pore entry pressure	0
Throat volume	0
η (Pore late filling)	1
S_w^*	0.84
P_c^* mode	Min
Multiphase conductance mode	Medium
Trapping mechanism	Activated

Table 9: Fluid properties implemented for sandstone 1 simulations

Parameter	**Value**
Mercury Interfacial Tension	485 mN/m
Mercury-Air Contact Angle	130 degrees
Water-Oil Interfacial Tension	30 mN/m
Water-Oil Contact Angle	8 degrees
Water Viscosity	1.05 cp
Oil Viscosity	1.39 cp

Capillary pressure and relative permeability results are shown in Figure 55 and Figure 56. It can be seen that the capillary pressure curve has a good match with experimental results in the lower mercury saturation. However, it starts to deviate at higher mercury saturations. This behavior was observed and explained in validation samples as well. The relative permeability curve was fitted to a Corey relative permeability model with the Corey exponent of 4 for water and a Corey exponent of 2 for oil. The fit is better in this case for water, but for oil, the curve of the simulation results is different in the middle water saturations.

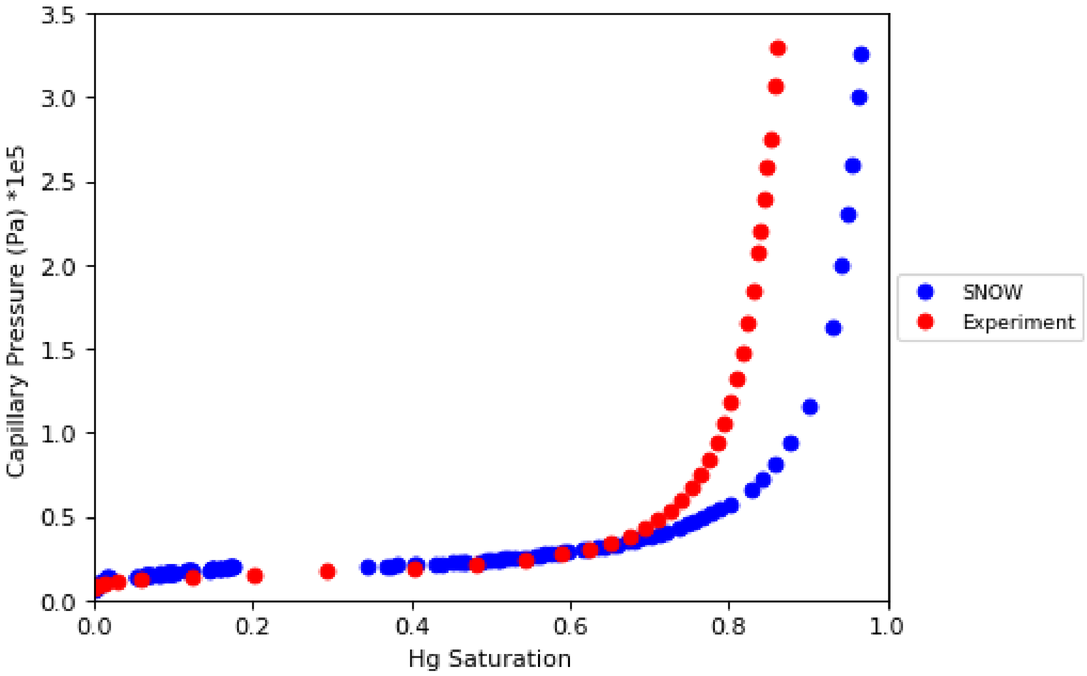


Figure 55: Capillary pressure curve of sandstone 1, comparing SNOW simulation with experimental results

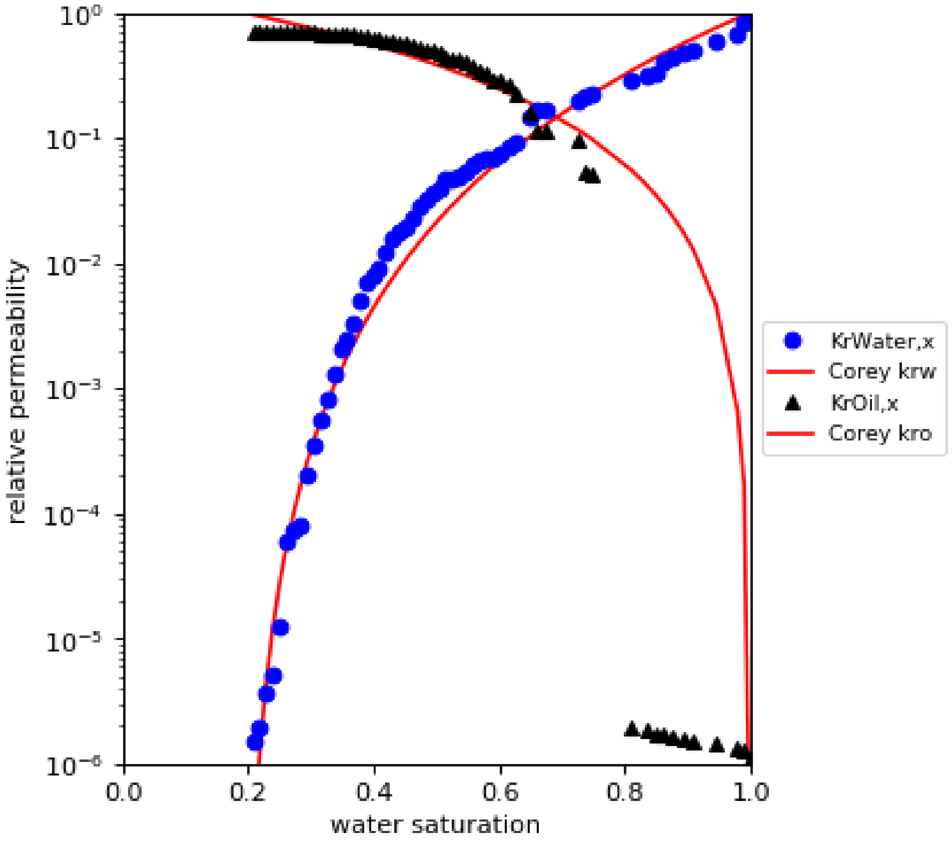


Figure 56: SNOW simulation results of relative permeability for sandstone 1, fitted with Corey relative permeabilities

5.2 Discussion

To derive the physical parameters of a rock sample using pore network modeling, we have started from the segmented micro-CT image of the rock. The segmented micro-CT image is input into an algorithm called SNOW to segment the pore space further into different regions corresponding to the pores of the network. After deriving the geometrical parameters of the network, physical formulas and phase properties are assigned to the network to calculate the desired physical properties. The simulations are in the end performed using invasion percolation theory and solving the mass balance over each pore to derive capillary pressure curves, absolute permeabilities, and relative permeabilities. The simulations were based only on the primary drainage process in which a nonwetting phase invades the pore network.

The conducted model was then validated using Berea sandstone, and Bentheimer sandstone experimental data found in the literature. During this validation, the shortcomings and strengths of our modeling were observed. In terms of capillary pressure, the curves were fitting the experimental data in the lower nonwetting phase saturation. However, the simulation results deviated from experimental data in higher nonwetting phase saturation due to the limitations in the imaging resolution. In the case of absolute permeability, the simulation results were higher by a factor of two in the case of Berea sandstone and lower by a factor of two third in the case of Bentheimer sandstone. However, such differences in the values of absolute permeabilities are also observed in experimental data, from one sample to another. Relative permeability observations indicated that the model fails at the endpoint relative permeabilities. This behavior is due to having an unconnected phase before the breakthrough at lower phase saturations. Nevertheless, the relative permeabilities have a good match with experimental data after the breakthrough and higher phase saturations.

As another part of model validation, the modeling parameters which can cause uncertainty in the simulation results were identified. The network extraction parameters, namely Sigma, and R_max, demonstrated to have a strong influence on the relative permeability curve. Moreover, pore late filling parameters, which are η, the initial fraction, and the adaptation of the capillary pressure of the pores from the capillary pressure of throats, mainly had an influence on the capillary pressure curve. This influence was not limited to the capillary pressure curve, but the pore later filling model parameters also had an impact on the relative permeability curves as well. The uncertainty in the aforementioned parameters causes a range of uncertainty in the simulation results as well. Using a correct workflow, these uncertainties are minimized, and the best possible match with the experimental results is obtained.

Based on these observations, a workflow is presented in Figure 57 to simulate capillary pressure and relative permeability curves using pore network modeling in the OpenPNM package and using the SNOW network extraction algorithm.

This workflow begins by finding the fractal dimensions and initial fraction (S_w^*) of the rock image of interest. These parameters are mainly found from MICP experiments, but since the purpose of this workflow is to have a prediction of the capillary pressure curve before having experiments, other alternatives could be considered. (Krohn, 1988) presented a method to derive the fractal dimensions of different rock samples based on a log-log plot of SEM images with different resolutions. (Peng et al., 2011) used grayscale micro CT images and a box-counting method to identify the fractal dimensions. (Zhang et al., 2018) calculated fractal dimensions of Bentheimer sandstone from petrophysical analysis using NMR and SIP measurements. The initial fraction (S_w^*) can be identified from the fraction of the pore space, which is missing from the pore or throat size distribution because of the image resolution limitations.

Another method to define the pore late filling model parameters is to calculate the pore volumes using the pore diameters. Then the initial fraction is defined as the fraction of the pore volume captured by this method over the voxel count pore space. In this method, η is one since we are considering spherical pores, and the P_c^* mode of 'min' is recommended for this implementation.

The second step in the workflow is to identify the correct sigma and R max parameters for SNOW extraction. Since relative permeability curves are found to be the most sensitive to the network structure, this can be done by simulating the relative permeability curves for different extraction parameters and choosing the ones that have the least artifacts in the relative permeability curves. After choosing the correct sigma and R max parameters, the absolute permeability and capillary pressure curves are calculated for the network.

Later on, in case of the availability of experimental data, the model is updated to match the experimental results by updating the parameters, e.g., P_c^* mode, η, S_w^*, Sigma, and R max.

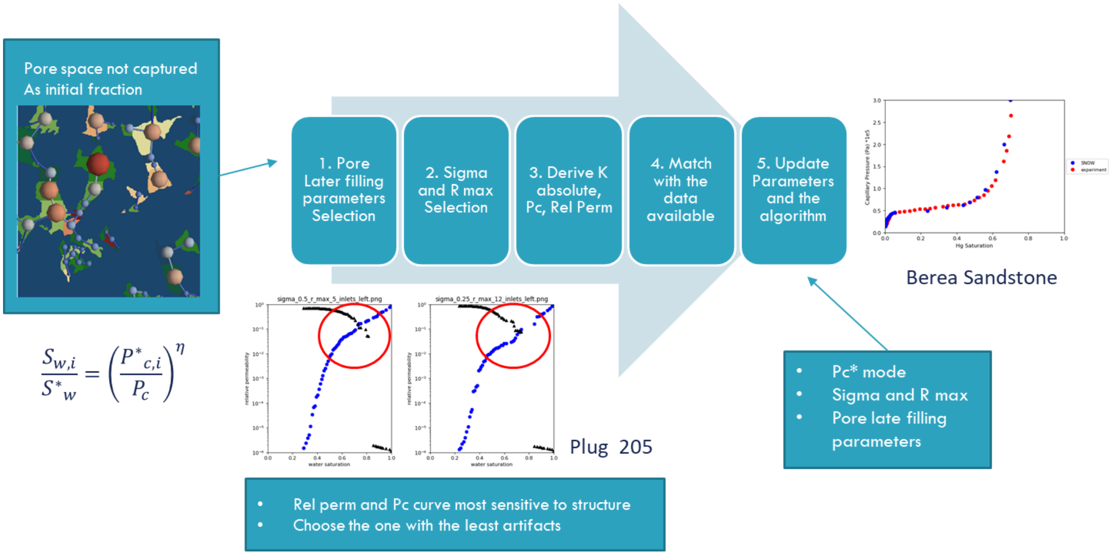

Pore space not captured
As initial fraction
$\frac{S_{w,i}}{S^*_w} = \left(\frac{P^*_{c,i}}{P_c}\right)^\eta$
1. Pore Later filling parameters Selection
2. Sigma and R max Selection
3. Derive K absolute, Pc, Rel Perm
4. Match with the data available
5. Update Parameters and the algorithm
Berea Sandstone
Hg Saturation
SNOW
experiment
sigma_0.5_r_max_5_inlets_left.png
sigma_0.25_r_max_12_inlets_left.png
water saturation
relative permeability
Plug 205
Pc* mode
Sigma and R max
Pore late filling parameters
Rel perm and Pc curve most sensitive to structure
Choose the one with the least artifacts

Figure 57: Workflow to simulate capillary pressure and relative permeability

Based on our observations on the investigated sandstones, the proper value for Sigma is around 0.5, and R_max is between 5 and 12. Based on the workflow presented in Figure 57, one can simulate capillary pressure and relative permeability of rock even when there is not a lot of data available from the rock. However, the simulation results suffer in this case in relative permeability endpoints and at higher nonwetting phase saturations for the capillary pressure curve.

Our simulations were only conducted for primary drainage of sandstones, and it can fail in more complex transport processes, e.g., imbibition due to more complex transport phenomena. Moreover, in the case of other rock types, e.g., carbonates, it can result in more inaccuracies in the results due to the presence of micropores in the rock, which are not implemented in our modeling. Further comprehensive modeling is required as a further study to include these complexities.

Conclusion

The pore network modeling using OpenPNM demonstrated the ability to predict transport properties and match the experimental data. However, the predictions possess inaccuracies in parts of its results. Based on the observations on the validation models and the sensitivity analysis, the pore late filling input parameters and their adaptation from the throats to the pores of the network has a great influence on the calculated capillary pressure curves. The changes in the capillary pressure curve also affects the relative permeability curves along with the SNOW extraction parameters which are R max and sigma.

A workflow is presented to understand the uncertainties in the modeling parameters and derive the most accurate results possible. A further study on more validation samples should be conducted to improve this workflow and find correlations between different rock types and uncertain parameters. The simulations and the workflow are based on the SNOW network extraction algorithm and OpenPNM pore network modeling python package.

Pore network modeling using OpenPNM provides a fast and flexible opensource tool to predict two-phase flow parameters using micro-CT images. The tool demonstrated promising results with errors existing in absolute permeability, capillary pressure at high nonwetting phase saturations, and endpoint relative permeabilities. Since it is an opensource tool, it can be extended to more complicated transport processes like imbibition and dynamic pore network modeling to improve the results.

List of Figures

List of Tables

Nomenclature

G	Shape factor	[]
A	Area	[m^2]
P	Perimeter	[m]
X	Coordinates of a voxel in the X-axis	[m]
Y	Coordinates of a voxel in the Y-axis	[m]
Z	Coordinates of a voxel in the Z-axis	[m]
R_{sphere}	Radius of a solid sphere	[m]
P_c	Capillary pressure	[pa]
σ	Interfacial Tension	[N/m]
θ	Contact Angle	[Radians]
r	Pore radius	[m]
$N(r)$	Number of Units with radius r	[]
D_f	Fractal Dimensions	[]
V_{Hg}	Volume of Mercury	[m^3]
l	Capillary tube length	[m]
$S_{w,i}$	Wetting phase saturation	[]
$S^*{}_w$	Wetting phase initial fraction	[]
$P^*{}_c$	Capillary pressure at an initial fraction of wetting phase saturation	[pa]

η	Fitting parameter corresponding to fractal dimensions	[]
q_i	Volumetric flow rate of pore i	[m^3/s]
$g_{h,ij}$	Hydraulic conductance of a throat connecting pore i to pore j	[m^3/pa.s]
P_i	Pressure of pore i	[pa]
P_j	Pressure of pore j	[pa]
μ	Fluid viscosity	[pa.s]
k	Hydraulic conductance	[m^3/pa.s]
SF_{pi}	Shape factor of pore i	[]
SF_t	Shape factor of a throat between pore i and j	[]
SF_{pj}	Shape factor of pore j	[]
D	Throat diameter	[m]
Q	Volumetric flow rate	[m^3/s]
K	Absolute permeability	[m^2]
L	Porous medium length	[m]
P_{in}	Inlet pressure	[pa]
P_{out}	Outlet pressure	[pa]
$K_{eff,p}$	Effective permeability of phase p	[m^2]
S_p	Saturation of phase p	[]
$K_{r,p}$	Relative permeability of phase p	[m^2]
N_p	Number of pores	[]

N_t	Number of throats	[]

Abbreviations

PNM	Pore Network Modeling
DNS	Direct Numerical Simulation
SNOW	Subnet of the Oversegmented Watershed
CT	Computer Tomography
SF	Shape Factor
DRP	Digital Rock Physics
MICP	Mercury Intrusion Porosimetry

References

Adler, P. (1990). Flow in simulated porous media. International Journal of Multiphase Flow, 16(4), 691-712.

Akbari et al. (2011). Viscous flow in variable cross-section microchannels of arbitrary shapes. International Journal of Heat and Mass Transfer, 54(17–18), 3970-3978.

Al-Raoush, R. (2005). Extraction of physically realistic pore network properties from three-dimensional synchrotron X-ray microtomography images of unconsolidated porous media systems. Journal of Hydrology, 300(1-4), 44-64.

Bakke, S., & Øren, P.-E. (1997). 3-D Pore-Scale Modelling of Sandstones and Flow Simulations in the Pore Networks. SPE 35479.

Baychev, T. e. (2019). Reliability of Algorithms Interpreting Topological and Geometric Properties of Porous Media for Pore Network Modelling. Transp Porous Media, 128, 271–301.

Bhattad et al. (2011). Effect of Network Structure on Characterization and Flow Modeling Using X-ray Micro-Tomography Images of Granular and Fibrous Porous Media. Transport in Porous Media, 90(363).

Biswal et al., B. (1999). Quantitative analysis of experimental and synthetic microstructures for sedimentary rock. Physica A: Statistical Mechanics and its Applications, 273(3-4), 452-475.

Bryant et al. (1993a). Network model evaluation of permeability and spatial correlation in a real random sphere packing. Transport in Porous Media, 11(1), 53-70.

Bryant, S. (1993b). Physically representative network models of transport in porous media. AIChE Journal, 39(3), 387-396.

Bryant, S., & Blunt, M. (1992). Prediction of relative permeability in simple porous media. Physical Review A, 46(4).

Chang et al. (2002). Magnetization Evolution in Network Models of Porous Rock under Conditions of Drainage and Imbibition. Journal of Colloid and Interface Science, 253(1), 159-170.

Churcher et. al. (1991). Rock Properties of Berea Sandstone, Baker Dolomite, and Indiana Limestone. Society of Petroleum Engineers.

Dathe et al. (2003). Dependence of the surface fractal dimension of soil pores on image resolution andmagnification. European Journal of Soil Science, 54, 453–466.

Dong, H., & Blunt, M. (2009). Pore-network extraction from micro-computerized-tomography images. Physical Review, 80(3).

Finney, J. (1970). Random packings and the structure of simple liquids. I. The geometry of random close packing. Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences, 319(1539).

Geodict. (2020). Geodict. Retrieved 2020, from geodict.de

Gostick, J. (2008). Multiphase Mass Transfer and Capillary Properties of Gas Diffusion Layers for Polymer Electrolyte Membrane Fuel Cells. PhD thesis presented to the University of Waterloo.

Gostick, J. (2008). Multiphase Mass Transfer and Capillary Properties of Gas Diffusion Layers for Polymer Electrolyte Membrane Fuel Cells.

Gostick, J. A. (2016). OpenPNM: A Pore Network Modeling Package.

Huang et at. (2019). From rock microstructure to macromechanical properties based on fractal dimensions. Advances in Mechanical Engineering.

J. Blunt, M. (2017). Multiphase Flow in Permeable Media: A Pore-Scale Perspective. Cambridge University Press.

J. Gostick et al. (2017). Versatile and efficient pore network extraction method using marker-based watershed segmentation. Physical Review E, 96(2).

Jeff T. Gostick et al. (2007). Pore network modeling of fibrous gas diffusion layers for polymer electrolyte membrane fuel cells. Power Sources, 173(1), 277-290.

Krohn, C. E. (1988). Fractal measurements of sandstones, shales, and carbonates. JOURNAL OF GEOPHYSICAL RESEARCH, 93(B4), 3297-3305.

Li, K. (2010). Analytical derivation of Brooks–Corey type capillary pressure models using fractal geometry and evaluation of rock heterogeneity. Journal of Petroleum Science and Engineering, 73, 20-26.

Linden, S., & Planas, B. (2019). FlowDict User Guide.

Lindquist et al. (1996). Medial axis analysis of void structure in three-dimensional tomographic images of porous media. Journal of Geophysical Research: Solid Earth, 101(B4), 8297–8310.

London, I. C. (n.d.). Berea Sandstone. Retrieved 5 5, 2020, from https://www.imperial.ac.uk/earth-science/research/research-groups/perm/research/pore-scale-modelling/micro-ct-images-and-networks/berea-sandstone/

Mason, G., & Marrow, N. (1991). Capillary behavior of a perfectly wetting liquid in irregular triangular tubes. Journal of Colloid and Interface Science, 141 (1), 262-274.

Masson, Y. (2016). A fast two-step algorithm for invasion percolation with trapping. Computers & Geosciences, 90(Part A), 41-48.

Muljadi et al. (2015). The impact of porous media heterogeneity on non-Darcy flow behaviour from pore-scale simulation. Advances in Water Resources, 95, 329-340.

Muljadi, B. P. (n.d.). digitalrocksportal. Retrieved May 6, 2020, from https://www.digitalrocksportal.org/projects/11/origin_data/17/

Oak, M. (1990). Three-Phase Relative Permeability of Water-Wet Berea. Society of Petroleum Engineers.

Oren et al. (1998). Extending Predictive Capabilities to Network Models. Society of Petroleum Engineers, 3(04).

Øren et al. (1998). Extending Predictive Capabilities to Network Models. SPE Journal, 3(4), 324-336.

Øren et at. (2007). Direct pore-scale computation of material and transport properties for North Sea reservoir rocks. Water Resources Research.

Peng et al. (2011). Computation of fractal dimension of rock pores based on gray CT images. Chin. Sci. Bull., 59(3346).

Planas, B., & Rief, S. (2019). PoroDict User Guide.

Prodanovi´c et al. (2007). 3D image-based characterization of fluid displacement in a Berea core. Advances in Water Resources, 30(2), 214–226.

Prodanovi´c et al. (2015). Imaged-based multiscale network modelling of microporosity in carbonates. Geological Society, London, Special Publications, 406(1), 95-113.

Rabbani et al. (2014). An automated simple algorithm for realistic pore network extraction from micro-tomography images. Journal of Petroleum Science and Engineering, 123, 164-171.

Ramstad et al. (2012). Relative permeability calculations from two-phase flow simulations directly on digital images of porous rocks. Transport in Porous Media.

Sahouli et al. (1999). Fractal Analysis of Mercury Porosimetry Data in the Framework of the Thermodynamic Method. Journal of Colloid and Interface Science, 214(2), 450-454.

Silin, D., & Patzek, T. (2006). Pore space morphology analysis using maximal inscribed spheres. Physica A: Statistical Mechanics and Its Applications, 371(2), 336–360.

Spanne et al. (1994). Synchrotron computed microtomograhy of porous media: Topology and transports. Physical Review Letters, 73(14), 2001-2004.

Swanberg, C. (2012). Characterizing the evolution of porosity and permeability in porous media undergoing pressure solution creep. Massachusetts Institute of Technology. Department of Earth, Atmospheric, and Planetary Sciences.

Taylor et al. (2015). A new method to identify void constrictions in micro-CT images of sand. Computers and Geotechnics, 69, 279-290.

Thovert et al. (1993). Computerized characterization of the geometry of real porous media: Their discretization, analysis and interpretation. Journal of Microscopy, 170(1), 65-79.

Tranter et al. (2007). Capillary Hysteresis in Neutrally Wettable Fibrous Media: A Pore Network Study of a Fuel Cell Electrode. Transp Porous Med, 121, 597–620.

Tsakiroglou et al. (1993). Pore-Wall Roughness as a Fractal Surface and Theoretical Simulation of Mercury Intrusion/Retraction in Porous Media. .Journal of Colloid and Interface Science Volume, 159(2), 287-301.

Tsakiroglou, , C. (1993). Pore-Wall Roughness as a Fractal Surface and Theoretical Simulation of Mercury Intrusion/Retraction in Porous Media. Journal of Colloid and Interface Science, 159(2), 287-301.

Valvatne, H. (2004). Predictive pore-scale modeling of two-phase flow in mixed wet media. Water Resour. Res., 40(7).

Washburn, E. (1921). The Dynamics of Capillary Flow. Phys. Rev., 17(273).

Wildenschild, D., & Sheppard, A. (2013). X-ray imaging and analysis techniques for quantifying pore-scale structure and processes in subsurface porous medium systems. Advances in Water Resources, 51, 217-246.

Y.-H. Lee et al. (1990). The fractal dimension as a measure of the roughness of rock discontinuity profiles. International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts,, 27(6), 453-464.

Yang et al. (2015). Extraction of pore-morphology and capillary pressure curves of porous media from synchrotron-based tomography data. Scientific Reports, 5, 10635.

Zhang et al. (2018). Enhanced pore space analysis by use of μ-CT, MIP, NMR, and SIP. Solid Earth, 9, 1225–1238.

Appendix A

OpenPNM Simulation Codes

A.1 SNOW network extraction

```
import porespy as ps
import matplotlib.pyplot as plt
import openpnm as op
import numpy as np
import scipy as sp
from porespy.networks import regions_to_network, add_boundary_regions
from porespy.networks import _net_dict
from porespy.networks import label_boundary_cells
from porespy.tools import pad_faces
from porespy.tools import make_contiguous
from porespy.metrics import region_surface_areas, region_interface_areas
from collections import namedtuple
import scipy.ndimage as spim
from porespy.filters import find_peaks, trim_saddle_points, trim_nearby_peaks
from skimage.morphology import watershed
from porespy.tools import randomize_colors

for sigma in [0.6]:
for r_max in [10]:
def snow_partitioning(im, dt=None, r_max=4, sigma=0.4, return_all=False,
mask=True, randomize=True):
tup = namedtuple('results', field_names=['im', 'dt', 'peaks', 'regions'])
print('_'*60)
print("Beginning SNOW Algorithm")
im_shape = sp.array(im.shape)
if im.dtype is not bool:
print('Converting supplied image (im) to boolean')
im = im > 0
if dt is None:
print('Peforming Distance Transform')
if sp.any(im_shape == 1):
ax = sp.where(im_shape == 1)[0][0]
dt = spim.distance_transform_edt(input=im.squeeze())
dt = sp.expand_dims(dt, ax)
```

```
else:
dt = spim.distance_transform_edt(input=im)

tup.im = im
tup.dt = dt

if sigma > 0:
print('Applying Gaussian blur with sigma =', str(sigma))
dt = spim.gaussian_filter(input=dt, sigma=sigma)

peaks = find_peaks(dt=dt, r_max=r_max)
print('Initial number of peaks: ', spim.label(peaks)[1])
peaks = trim_saddle_points(peaks=peaks, dt=dt, max_iters=500)
print('Peaks after trimming saddle points: ', spim.label(peaks)[1])
peaks = trim_nearby_peaks(peaks=peaks, dt=dt)
peaks, N = spim.label(peaks)
print('Peaks after trimming nearby peaks: ', N)
tup.peaks = peaks
if mask:
mask_solid = im > 0
else:
mask_solid = None
regions = watershed(image=-dt, markers=peaks, mask=mask_solid)
if randomize:
regions = randomize_colors(regions)
if return_all:
tup.regions = regions
return tup
else:
return regions

def snow(im, voxel_size=1,
boundary_faces=['top', 'bottom', 'left', 'right', 'front', 'back'],
marching_cubes_area=False):

# -----------------------------------------------------------------------
# SNOW void phase
regions = snow_partitioning(im=im,  r_max=r_max, sigma=sigma, return_all=True)
im = regions.im
dt = regions.dt
regions = regions.regions
b_num = sp.amax(regions)
# -----------------------------------------------------------------------
# Boundary Conditions
regions = add_boundary_regions(regions=regions, faces=boundary_faces)
# -----------------------------------------------------------------------
# Padding distance transform and image to extract geometrical properties
dt = pad_faces(im=dt, faces=boundary_faces)
im = pad_faces(im=im, faces=boundary_faces)
regions = regions*im
regions = make_contiguous(regions)
# -----------------------------------------------------------------------
# Extract void and throat information from image
net = regions_to_network(im=regions, dt=dt, voxel_size=voxel_size)
# -----------------------------------------------------------------------
# Extract marching cube surface area and interfacial area of regions
```

```
if marching_cubes_area:
areas = region_surface_areas(regions=regions)
interface_area = region_interface_areas(regions=regions, areas=areas,
voxel_size=voxel_size)
net['pore.surface_area'] = areas * voxel_size**2
net['throat.area'] = interface_area.area
# ------------------------------------------------------------------------
# Find void to void connections of boundary and internal voids
boundary_labels = net['pore.label'] > b_num
loc1 = net['throat.conns'][:, 0] < b_num
loc2 = net['throat.conns'][:, 1] >= b_num
pore_labels = net['pore.label'] <= b_num
loc3 = net['throat.conns'][:, 0] < b_num
loc4 = net['throat.conns'][:, 1] < b_num
net['pore.boundary'] = boundary_labels
net['throat.boundary'] = loc1 * loc2
net['pore.internal'] = pore_labels
net['throat.internal'] = loc3 * loc4
# ------------------------------------------------------------------------
# label boundary cells
net = label_boundary_cells(network=net, boundary_faces=boundary_faces)
# ------------------------------------------------------------------------
# assign out values to dummy dict

temp = _net_dict(net)
temp.im = im.copy()
temp.dt = dt
temp.regions = regions
return temp

ws = op.Workspace()
ws.clear()
ws.keys()
proj = ws.new_project()
from skimage import io
im = io.imread('Berea.tif')
imtype=im.view()
digits=np.prod(np.array(im.shape))
logi=(np.sum(im==0)+np.sum(im==1))==digits
imtype=im.view()
im = sp.array(im, dtype=bool)
im = ~im


net = snow(im, voxel_size=5.345e-6,
boundary_faces=['top', 'bottom', 'left', 'right', 'front', 'back'],
marching_cubes_area=False) # voxel size and marching cube can be changed for each specific sample
pn = op.network.GenericNetwork()
pn.update(net)


a = pn.check_network_health()
op.topotools.trim(network=pn,pores=a['trim_pores'])
```

```
105.    mgr=op.Workspace()
106.    mgr.save_workspace('Berea_r'+str(r_max)+'_sigma'+str(sigma)+'.pnm')
```

The presented code is used to extract a pore network from the Berea sandstone image using the SNOW extraction method. The required Python packages are imported until line 16, and then the intended sigma and r_max extraction parameters are defined in lines 17 and 18. Please note that a list of different sigma and r_max can be defined here to have different networks extracted. Line 19 till 42 marks a function that receives this sigma and r_max parameter and then segments the pore space image into different regions using the SNOW algorithm. Lines 43 till 87 are a function that receives this segmented image, and after adding boundary conditions and assigning geometrical parameters, it makes a pore network that is readable by the OpenPNM package.

In lines 88 to 91, we initialize an OpenPNM project and then read the binary image inline 93. Lines 94 till 98 checks the quality of the image, and line 99 is necessary so that the extracted network matches the initial binary image. The SNOW extraction functions are called in code line 100 and assigned to the OpenPNM project in code line 102. The isolated pores are detected in code line 103 and removed by line 104. Then we can save this extracted network with lines 105 and 106 and then import it into another code for further simulations.

A.2 Capillary pressure simulations of Berea Sandstone

```
1.      import openpnm as op
2.      import numpy as np

3.      inlets = 'left'

4.      import matplotlib.pyplot as plt
5.      from matplotlib.font_manager import FontProperties
6.      #setting up subplots
7.      fig = plt.figure(figsize=(6, 10), dpi=80, facecolor='w', edgecolor='k')
8.      ax = fig.add_subplot(211)   #top
9.      ax.set_ylim([0,3])
10.     ax.set_xlim([0,1])
11.     ax.set_xlabel('Hg Saturation')
12.     ax.set_ylabel('Capillary Pressure (Pa) *1e5')

13.     fontP = FontProperties()
14.     fontP.set_size('small')

15.     j = 0
16.     sat_save=np.zeros((100,60))
17.     sat_save2=np.zeros((100,60))
```

```
for eta in [0.85]:
for sigma in [0.6]:
for r_max in [10]:
print('*'*60)
print('eta: ', eta, ' sigma: ', sigma, 'r_max: ', r_max)
mgr = op.Workspace()
mgr.clear()
mgr.keys()
mgr.load_workspace('Berea_r'+str(r_max)+'_sigma'+str(sigma)+'.pnm')
pn=mgr['sim_02']['net_01']
project = pn.project

del pn['pore.area']
del pn['throat.conduit_lengths.pore1']
del pn['throat.conduit_lengths.pore2']
del pn['throat.conduit_lengths.throat']
del pn['throat.endpoints.tail']
del pn['throat.endpoints.head']
del pn['throat.volume']
del pn['pore.volume']


pn['throat.volume'] = 0
pn.add_model(propname='throat.endpoints',
model=op.models.geometry.throat_endpoints.spherical_pores)
pn.add_model(propname='pore.area',
model=op.models.geometry.pore_area.sphere)
pn.add_model(propname='pore.volume',
model=op.models.geometry.pore_volume.sphere)
pn.add_model(propname='throat.conduit_lengths',
model=op.models.geometry.throat_length.conduit_lengths)
pn.add_model(propname='pore.volume',
model=op.models.geometry.pore_volume.sphere)


geom = op.geometry.GenericGeometry(network=pn)
geom['pore.volume'][pn.pores('left')] = 0
hg = op.phases.GenericPhase(network=pn)
hg['pore.surface_tension'] = 480e-3
hg['pore.contact_angle'] = 140
hg['throat.surface_tension'] = 480e-3
hg['throat.contact_angle'] = 140
phys = op.physics. GenericPhysics (network=pn, phase=hg, geometry=geom)
phys['pore.entry_pressure'] = 0

phys.add_model(propname='throat.entry_pressure',
model=op.models.physics.capillary_pressure.washburn)
phys.add_model(propname='pore.pc_star',
model=op.models.misc.from_neighbor_throats,
throat_prop='throat.entry_pressure',
mode='min')
phys.add_model(propname='pore.late_filling',
model=op.models.physics.multiphase.late_filling,
pressure='pore.pressure',
Pc_star='pore.pc_star',
eta=eta, Swp_star=0.21,
v.      regen_mode='deferred')
55.     # phys['throat.pc_star'] = phys['throat.entry_pressure']
56.     # phys.add_model(propname='throat.late_filling',
57.     #       model=op.models.physics.multiphase.late_filling,
58.     #       pressure='throat.pressure',
59.     #       Pc_star='throat.pc_star',
60.     #       eta=eta, Swp_star=0.21,
61.     #       regen_mode='deferred')

62.     MP_1 = op.algorithms.MixedInvasionPercolation(network=pn)
63.     MP_1.setup(phase=hg, late_pore_filling='pore.late_filling')
64.     MP_1.set_inlets(pn.pores(inlets))
65.     MP_1.run()
66.     #MP_1.set_outlets(pn.pores(['right']))
67.     #MP_1.apply_trapping()
68.     #MP_1.plot_intrusion_curve(inv_points=range(40000,60000,100))

69.     data = MP_1.get_intrusion_data()
70.     data2 = MP_1.get_intrusion_data(inv_points=[2e5,2.5e5,2.75e5])
71.     for x in range(len(data.S_tot)):
a.      sat_save[x,j] = data.S_tot[x]
72.     for x in range(len(data2.S_tot)):
a.      sat_save2[x,j] = data2.S_tot[x]
73.     j=j+1

74.     p1, = ax.plot(data.S_tot, data.Pcap/1e5,'bo')
75.     p3, = ax.plot(data2.S_tot, np.asarray(data2.Pcap)/1e5,'bo')
76.     lgd1 = ax.legend([p1, p2],
i.      ["SNOW",
ii.     "Experiment"],
iii.    #'Geodict',
iv.     loc='center left', bbox_to_anchor=(1, 0.5), prop = fontP)

77.     #title ='sigma_'+ str(sigma)+ '_r_max_'+str(r_max)+'_inlets_' +str(inlets)+'.png'
78.     #ax.set_title(title)
79.     #fig.savefig(title)

80.     plt.show()
81.     #op.io.VTK.save(network=pn, filename='Berea_r'+str(r_max)+'_sigma'+str(sigma))
```

In code lines 1 to 14, the necessary packages are imported, and the initial setup is done for further plotting through the program. Lines 15 to 17 define variables that are later implemented for saving parameters and further export from the program. Line 18 marks the beginning of a loop to define different etas in the pore late filling model. This can be changed to S_w^* also, but the variable must be set in line 54 of the program. Lines 19 and 20 mark the loops for importing different sets of pore networks with different sigma and different r_max. Here a list of parameters can be set for the loops. Please note the indentation required for each loop.

Lines 21 to 28 mark the initialization of the project and clear any existing workspace. Since our networks are extracted using the SNOW extraction algorithm, we delete some of the parameters that we would like to redefine through the program in lines 29 to 36. This redefinition is to make a model that complies with the assumptions of the physical formulas in our modeling. From lines 37 to 42, we redefine the properties of our network.

In code line 43, we assign a generic geometry to our model and then set our inlets volumes to zero in code line 44. Then in code line 45, we define our phase as mercury and assign its properties until line 49.

In code line 50, we define a generic physics for our modeling and set pore entry pressure to zero in code line 51. This is because we want our mixed invasion percolation algorithm to act as an invasion percolation for our drainage simulations.

From lines 52 to 61, we define our pore late filling model. Lines 55 till 61 are commented out since we defined the throat volumes as zero, and therefore the pore late filling model of the throats will not affect our simulations. Line 53 marks the adaptation of the pc_star from the throats connected to a pore, and line 54 defines the pore late filling model based on that.

Lines 62 to 68 represent the implementation of the mixed percolation algorithm to run capillary pressure simulations. The phase and the pore late filling model are assigned to the algorithm set up in code line 63, and the inlets are set at line 64. Lines 66 to 68 are optional in case of interest in using the trapping mechanism and plotting invasion points manually. Lines 69 to 73 receives the simulation data and saves them in corresponding variables.

Lines 74 till 79 are used for plotting the results, and using line 81; we can export the results to VTK format, which is readable by Paraview software.

A.3 Permeability Simulation Code for Berea Sandstone

```
1.    import openpnm as op
2.    import numpy as np

3.    inlets='left'
4.    outlets= 'right'

5.    j = 0
6.    sat_save=np.zeros((100,60))
7.    kro_save=np.zeros((100,60))
8.    krw_save=np.zeros((100,60))

9.    for eta in [0.85]:
10.   for sigma in [0.6]:
11.   for r_max in [10]:
12.   print('*'*60)
13.   print('eta: ', eta, ' sigma: ', sigma, 'r_max: ', r_max)
14.   mgr = op.Workspace()
15.   mgr.clear()
16.   mgr.keys()
```

```
mgr.load_workspace('Berea_r'+str(r_max)+'_sigma'+str(sigma)+'.pnm')
pn=mgr['sim_02']['net_01']
project = pn.project

del pn['pore.area']
del pn['throat.conduit_lengths.pore1']
del pn['throat.conduit_lengths.pore2']
del pn['throat.conduit_lengths.throat']
del pn['throat.endpoints.tail']
del pn['throat.endpoints.head']
del pn['throat.volume']

pn['throat.volume'] = 0
pn['throat.diameter']=pn['throat.equivalent_diameter']

pn.add_model(propname='throat.endpoints',
model=op.models.geometry.throat_endpoints.spherical_pores)
pn.add_model(propname='pore.area',
model=op.models.geometry.pore_area.sphere)
pn.add_model(propname='pore.volume',
model=op.models.geometry.pore_volume.sphere)
pn.add_model(propname='throat.conduit_lengths',
model=op.models.geometry.throat_length.conduit_lengths)

geom = op.geometry.GenericGeometry(network=pn)

hg = op.phases.Mercury(network=pn)
phys = op.physics.Standard(network=pn, phase=hg, geometry=geom)
phys['pore.entry_pressure'] = 0.0
MP_0= op.algorithms.MixedInvasionPercolation(project=project)
MP_0.setup(phase=hg)
MP_0.set_inlets(pores=pn.pores(['left']))
MP_0.run()
MP_0.set_outlets(pn.pores(['right']))
MP_0.apply_trapping()
#MP_0.plot_intrusion_curve()
data0 = MP_0.get_intrusion_data()
max_pc = data0.Pcap[-1]
hg['pore.occupancy'] = MP_0.resultsOmid(Pc=max_pc)['pore.occupancy']
#water['pore.occupancy'] = ~oil['pore.occupancy']
residual_sat = sum(pn['pore.volume'][~hg['pore.occupancy']])/sum(pn['pore.volume'])
print('residual saturation:'); print(residual_sat)

water = op.phases.GenericPhase(network=pn)
water['throat.contact_angle'] = 172
water['pore.contact_angle'] = 172
water['throat.surface_tension'] = 30e-3
water['pore.surface_tension'] = 30e-3
water['pore.viscosity']=1.05e-3

phys_water = op.physics.GenericPhysics(network=pn, phase=water, geometry=geom)
phys_water.add_model(propname='throat.flow_shape_factors',
model=op.models.physics.flow_shape_factors.ball_and_stick)
phys_water.add_model(propname='throat.hydraulic_conductance',
model=op.models.physics.hydraulic_conductance.hagen_poiseuille)
phys_water.add_model(propname='throat.entry_pressure',
model=op.models.physics.capillary_pressure.washburn,
diameter='throat.diameter')
phys_water['pore.entry_pressure'] = 0.0
phys_water.add_model(propname='pore.pc_star',
model=op.models.misc.from_neighbor_throats,
throat_prop='throat.entry_pressure',
mode='max')
phys_water.add_model(propname='pore.late_filling',
model=op.models.physics.multiphase.late_filling,
pressure='pore.pressure',
Pc_star='pore.pc_star',
eta=0.85, Swp_star=0.21,
regen_mode='deferred')
phys_water['throat.pc_star'] = phys_water['throat.entry_pressure']
phys_water.add_model(propname='throat.late_filling',
model=op.models.physics.multiphase.late_filling,
pressure='throat.pressure',
Pc_star='throat.pc_star',
eta=0.85, Swp_star=0.21,
regen_mode='deferred')

oil = op.phases.GenericPhase(network=pn)
oil['pore.viscosity']=1.39e-3
oil['throat.viscosity']=1.39e-3
oil['throat.contact_angle'] =98
oil['pore.contact_angle']=98
oil['throat.surface_tension'] = 30e-3
oil['pore.surface_tension']=30e-3
phys_oil = op.physics.GenericPhysics(network=pn, phase=oil, geometry=geom)
phys_oil.add_model(propname='throat.flow_shape_factors',
model=op.models.physics.flow_shape_factors.ball_and_stick)
phys_oil.add_model(propname='throat.hydraulic_conductance',
model=op.models.physics.hydraulic_conductance.hagen_poiseuille)
phys_oil.add_model(propname='throat.entry_pressure',
model=op.models.physics.capillary_pressure.washburn,
diameter='throat.diameter')
phys_oil['pore.entry_pressure'] = 0.0
phys_oil.add_model(propname='pore.pc_star',
model=op.models.misc.from_neighbor_throats,
throat_prop='throat.entry_pressure',
mode='min') #important to change
phys_oil.add_model(propname='pore.late_filling',
model=op.models.physics.multiphase.late_filling,
pressure='pore.pressure',
Pc_star='pore.pc_star',
eta=eta, Swp_star=0.21,
regen_mode='deferred')
phys_oil['throat.pc_star'] = phys_oil['throat.entry_pressure']
phys_oil.add_model(propname='throat.late_filling',
model=op.models.physics.multiphase.late_filling,
pressure='throat.pressure',
Pc_star='throat.pc_star',
eta=0.85, Swp_star=0.21,
regen_mode='deferred')
```

```
flow = op.algorithms.StokesFlow(network=pn, phase=oil)
flow.setup(conductance='throat.hydraulic_conductance')
flow.set_value_BC(pores=pn.pores('left'), values=2e5)
flow.set_value_BC(pores=pn.pores('right'), values=1e5)
flow.run()
area = (400*5.345e-6)**2
length = 400*5.345e-6
permeability = flow.calc_effective_permeability(domain_area=area,
domain_length=length,
inlets=pn.pores('left'),
outlets=pn.pores('right'))
project.purge_object(obj=flow)
print('permeability is:'); print(permeability/9.869233e-13)


MP_1 = op.algorithms.MixedInvasionPercolation(network=pn)
MP_1.setup(phase=oil, late_pore_filling='pore.late_filling')
MP_1.set_inlets(pn.pores(inlets))
MP_1.run()
MP_1.set_outlets(pn.pores([outlets]))
MP_1.apply_trapping()
#MP_1.plot_intrusion_curve()


data = MP_1.get_intrusion_data()#(inv_points=range(0,4000))
# Filter for evenly spaced sat inc. first and last
filter_pc = [data.Pcap[0]]
sat = [data.S_tot[0]]
for i, pc in enumerate(data.Pcap):
if  data.S_tot[i] - sat[-1] > 0.01:
filter_pc.append(pc)
sat.append(data.S_tot[i])
filter_pc.append(data.Pcap[-1])
sat.append(data.S_tot[-1])


def update_phase_and_phys(results):
oil['pore.occupancy'] = results['pore.occupancy']
water['pore.occupancy'] = ~results['pore.occupancy']
oil['throat.occupancy'] = results['throat.occupancy']
water['throat.occupancy'] = ~results['throat.occupancy']
#adding multiphase conductances
mode='medium' #important to change
phys_oil.add_model(model=op.models.physics.multiphase.conduit_conductance,
propname='throat.conduit_hydraulic_conductance',
throat_conductance='throat.hydraulic_conductance',
mode=mode)
phys_water.add_model(model=op.models.physics.multiphase.conduit_conductanc,
propname='throat.conduit_hydraulic_conductance',
throat_conductance='throat.hydraulic_conductance',
mode=mode)
rel_perm_oil = []
rel_perm_water = []

for Pc in filter_pc:
update_phase_and_phys(MP_1.resultsOmid(Pc=Pc))
```

```
# Multiphase
Stokes_alg_multi_phase_oil = op.algorithms.StokesFlow(network=pn,phase=oil)
Stokes_alg_multi_phase_oil.setup(conductance='throat.conduit_hydraulic_conductance')
Stokes_alg_multi_phase_water = op.algorithms.StokesFlow(network=pn,phase=water)
Stokes_alg_multi_phase_water.setup(conductance='throat.conduit_hydraulic_conductance')

#BC inlets
Stokes_alg_multi_phase_oil.set_value_BC(values=2e5, pores=pn.pores(['left']))
Stokes_alg_multi_phase_water.set_value_BC(values=2e5, pores=pn.pores(['left']))

#BC outlets
Stokes_alg_multi_phase_oil.set_value_BC(values=1e5, pores=pn.pores(['right']))
Stokes_alg_multi_phase_water.set_value_BC(values=1e5, pores=pn.pores(['right']))

# Run Multiphase algs
Stokes_alg_multi_phase_oil.run()
Stokes_alg_multi_phase_water.run()

effective_permeability_oil_multi = Stokes_alg_multi_phase_oil.calc_effective_permeability(domain_area=area,
domain_length=length,
inlets=pn.pores(['left']),
outlets=pn.pores(['right']))
effective_permeability_water_multi = Stokes_alg_multi_phase_water.calc_effective_permeability(domain_area=area,
domain_length=length,
inlets=pn.pores(['left']),
outlets=pn.pores(['right']))
Stokes_alg_multi_phase_water.rate
relative_eff_perm_oil = effective_permeability_oil_multi/permeability
relative_eff_perm_water = effective_permeability_water_multi/permeability

rel_perm_oil.append(relative_eff_perm_oil)
rel_perm_water.append(relative_eff_perm_water)

pn['pore.oil_occupacy@Pc='+str(Pc)] = oil['pore.occupancy']
pn['pore.water_occupacy@Pc='+str(Pc)] = water['pore.occupancy']

project.purge_object(obj=Stokes_alg_multi_phase_oil)
project.purge_object(obj=Stokes_alg_multi_phase_water)

# sat_save.append(1 - np.asarray(sat))
# kro_save.append(np.asarray(rel_perm_oil))
# krw_save.append(np.asarray(rel_perm_water))
#
==========================================================================================
# plotting
#
==========================================================================================

import matplotlib.pyplot as plt
from matplotlib.font_manager import FontProperties
'exec(%matplotlib inline)'

fontP = FontProperties()
fontP.set_size('small')
```

```
#setting up subplots
fig = plt.figure(figsize=(6, 10), dpi=80, facecolor='w', edgecolor='k')
ax1 = fig.add_subplot(211)   #top

oak_sat=[0.995,0.996,0.975,0.96,0.955,0.932,0.912,0.875,0.839,0.793,0.754,
0.716,0.671,0.623,0.583,0.548,0.511,0.48,0.446,0.42,0.391,0.361,0.33,
0.304,0.266,0.264,]
oak_kro=[1,0.992,0.973,0.867,0.896,0.802,0.616,0.489,0.378,0.278,0.203,0.135,
0.0898,0.0549,0.033,0.0205,0.0126,0.00763,0.00461,0.00276,0.00163,
0.000943,0.000519,0.00028,0,0,]
oak_krw=[0,0.000704,0.00138,0.00246,0.00509,0.00911,0.014,0.0222,0.0344,0.056,
0.0739,0.0982,0.131,0.16,0.192,0.239,0.292,0.355,0.429,0.514,0.607,
0.702,0.772,0.834,0.887,0.9,]

watersat = 1 - np.asarray(sat)
corey_krw = ((watersat-residual_sat)/(1-residual_sat))**4
corey_kro = ((1-watersat)/(1-residual_sat))**2.5

#plots for subplot1 - strict permeability
p2, = ax1.plot(oak_sat, oak_krw, 'ro')
p5, = ax1.plot(oak_sat, oak_kro, 'r^')
#p8, = ax1.plot(watersat, corey_kro, linestyle ='-' , color = 'r')
#p9, = ax1.plot(watersat, corey_krw, linestyle ='-' , color = 'r')
ax1.set_ylabel('relative permeability')
ax1.set_xlabel("water saturation")
ax1.set_ylim([1e-6,1])
ax1.set_xlim([0,1])
ax1.set_yscale('log')
title ='sigma_'+ str(sigma)+ '_r_max_'+str(r_max)+'_inlets_' +str(inlets)+'.png'
#ax1.set_title(title)

fig.subplots_adjust(left=0.13, right=.7, top=0.95, bottom=0.05)

sat = 1 - np.asarray(sat)
p1, = ax1.plot(sat, rel_perm_water, 'bo')
p4, = ax1.plot(sat, rel_perm_oil, 'k^')
for x in range(len(sat)):
sat_save[x,j] = sat[x]
kro_save[x,j] = rel_perm_oil[x]
krw_save[x,j] = rel_perm_water[x]
j=j+1
#need to work on legend to match up with the right things
lgd1 = ax1.legend([p1, p2, p4,p5],
["SNOW KrWater,x", "Experimental krw",
"SNOW KrOil,x", "Experimental kro"],
#'Geodict KrOil', 'Geodict KrWater'],
#'Corey Oil', 'Corey Water'],
loc='center left', bbox_to_anchor=(1, 0.5), prop = fontP)
#fig.savefig(title)
plt.show()
```

To run the permeability simulations, inlets and outlets are defined in lines 3 and 4, and then the variables to save the results of the simulations are defined in lines 5 to 8. The η parameter of the pore late filling model is defined in the first loop in code line 9. Sigma and R_max input parameters to construct the pore network are read in the second loop in code line 10 and the third loop in code line 11. Therefore, the corresponding network which was created using the SNOW extraction algorithm is then loaded using code lines 14 till 19. Some properties of the network are redefined the same as in capillary pressure simulations in lines 20 until line 32. Please note that as recommended by (J. Gostick et al., 2017), to simulate permeability data, equivalent throat diameters are used, which are assigned to the network in code line 28.

The code implemented in lines 34 to 49 can define the residual saturation of the network using the trapping mechanism. This algorithm can define the pores that are hydraulically bypassed in the mixed percolation algorithm. The volume summation of these pores is then divided into the whole pore volume to find the residual saturation.

Water properties and the physics of the phase are defined in lines 50 to 64. Please note that since in our drainage relative permeability simulations, we assume that the rock sample is initially water-saturated and then it is invaded by an oil phase, the pore late filling parameters of the water phase is not going to affect our results in such a simulation algorithm. The oil phase properties and physics are defined in lines 65 to 80 correspondingly.

The absolute permeability is defined using the code in lines 81 to 90, which uses single-phase stokes flow to derive an absolute permeability. It is then later used in the code to derive relative permeabilities.

The invasion of the oil phase into the water-filled sample is coded using a mixed percolation algorithm in lines 91 to 97, and the data are extracted in lines 98 and 99. These data are then spaced based on the spacing of the saturation points, having a distance of at least 0.01 through the code lines 100 to 104.

The function, which is defined in 105, defines the oil and water fluid occupancies using the results coming from mixed invasion percolation and then uses these occupancies to define new hydraulic conductance. Using this new conductance, the relative permeability of each saturation step can be defined. The mode of this conductance change, which can affect the results heavily, is defined in code line f. The mode ‘medium’ is found to be matching the experimental results the best way. Please note that the results section of the OpenPNM source code is modified for this function to work. This modification is further explained through this document.

The loop starting in line 108 calculates the relative permeabilities in different saturation steps, which are resulted from mixed invasion percolation. As can be seen in line a, there is a new class defined in OpenPNM source named “resultsOmid” and it is used as an input for the function in code line 105. The code for this new class is demonstrated at the end of this appendix. The main difference here with the main result function of the OpenPNM is in line 26 of the results code. In this code line, the multiplication with the volume is removed only to get the occupancy results.

After calculating the relative permeability results, the numbers can be saved into predefined variables for export as in line 109 till 111. In case this method of saving the results appear cumbersome to the reader, one can use the method of saving results in capillary pressure simulations.

The plotting of the results is done using the code lines 113 to 150. Here experimental results can also be added for the comparison purposes as in lines 123 to 125, in which (Oak, 1990) experimental results are used to compare with simulation results for Berea sandstone.

```
1.    def resultsOmid(self, Pc):
2.    r"""
3.    Places the results of the IP simulation into the Phase object.

4.    Parameters
5.    ----------
6.    Pc : float
7.    Capillary Pressure at which phase configuration was reached

8.    """
9.    if Pc is None:
10.   results = {'pore.invasion_sequence':
i.    self['pore.invasion_sequence'],
ii.   'throat.invasion_sequence':
iii.  self['throat.invasion_sequence']}
11.   else:
12.   phase = self.project.find_phase(self)
13.   net = self.project.network
14.   inv_p = self['pore.invasion_pressure'].copy()
15.   inv_t = self['throat.invasion_pressure'].copy()
16.   # Handle trapped pores and throats by moving their pressure up
17.   # to be ignored
18.   if np.sum(self['pore.invasion_sequence'] == -1) > 0:
a.    inv_p[self['pore.invasion_sequence'] == -1] = Pc + 1
19.   if np.sum(self['throat.invasion_sequence'] == -1) > 0:
a.    inv_t[self['throat.invasion_sequence'] == -1] = Pc + 1
20.   p_inv = inv_p <= Pc
21.   t_inv = inv_t <= Pc

22.   if self.settings['late_pore_filling']:
a.    # Set pressure on phase to current capillary pressure
b.    phase['pore.pressure'] = Pc
c.    # Regenerate corresponding physics model
d.    for phys in self.project.find_physics(phase=phase):
e.    phys.regenerate_models(self.settings['late_pore_filling'])
f.    # Fetch partial filling fraction from phase object (0->1)
g.    frac = phase[self.settings['late_pore_filling']]
```

```
p_vol = net['pore.volume']*frac
else:
p_vol = net['pore.volume']
if self.settings['late_throat_filling']:
# Set pressure on phase to current capillary pressure
phase['throat.pressure'] = Pc
# Regenerate corresponding physics model
for phys in self.project.find_physics(phase=phase):
phys.regenerate_models(self.settings['late_throat_filling'])
# Fetch partial filling fraction from phase object (0->1)
frac = phase[self.settings['late_throat_filling']]
t_vol = net['throat.volume']*frac
else:
t_vol = net['throat.volume']
results = {'pore.occupancy': p_inv,
'throat.occupancy': t_inv}
return results
```

A.4 Relative Permeability Calculations for the Contact Angle and IFT sensitivity analysis

The contact angle and IFT for the new relative permeability curves can be adjusted in lines two and three.

```
1.  for Pc in filter_pc:
2.                  contact_angle = 8
3.                  IFT = 30
4.                  factor = IFT/30 * math.cos(math.radians(180-contact_angle))/math.cos(math.radians(172))
5.                  Pc = Pc*factor
6.
7.                  update_phase_and_phys(MP_1.resultsOmid(Pc=Pc))
8.                  # Multiphase
9.                  Stokes_alg_multi_phase_oil = op.algorithms.StokesFlow(network=pn,phase=oil)
10.                 Stokes_alg_multi_phase_oil.setup(conductance='throat.conduit_hydraulic_conductance')
11.                 Stokes_alg_multi_phase_water = op.algorithms.StokesFlow(network=pn,phase=water)
12.                 Stokes_alg_multi_phase_water.setup(conductance='throat.conduit_hydraulic_conductance')
13.
14.                 #BC inlets
15.                 Stokes_alg_multi_phase_oil.set_value_BC(values=2e5, pores=pn.pores(['left']))
16.                 Stokes_alg_multi_phase_water.set_value_BC(values=2e5, pores=pn.pores(['left']))
17.
18.                 #BC outlets
19.                 Stokes_alg_multi_phase_oil.set_value_BC(values=1e5, pores=pn.pores(['right']))
20.                 Stokes_alg_multi_phase_water.set_value_BC(values=1e5, pores=pn.pores(['right']))
21.
22.                 # Run Multiphase algs
23.                 Stokes_alg_multi_phase_oil.run()
24.                 Stokes_alg_multi_phase_water.run()
25.
```

```
                effective_permeability_oil_multi = Stokes_alg_multi_phase_oil.calc_effective_permeability(domain_area=area,
                                                                                                          domain_length=length,
                                                                                                          inlets=pn.pores(['left']),
                                                                                                          outlets=pn.pores(['right']))
                effective_permeability_water_multi = Stokes_alg_multi_phase_water.calc_effective_permeability(domain_area=area,
                                                                                                              domain_length=length,
                                                                                                              inlets=pn.pores(['left']),
                                                                                                              outlets=pn.pores(['right']))
                Stokes_alg_multi_phase_water.rate
                relative_eff_perm_oil = effective_permeability_oil_multi/permeability
                relative_eff_perm_water = effective_permeability_water_multi/permeability
```

Appendix B

Relative Permeability Match with Corey's Model for Berea Sandstone

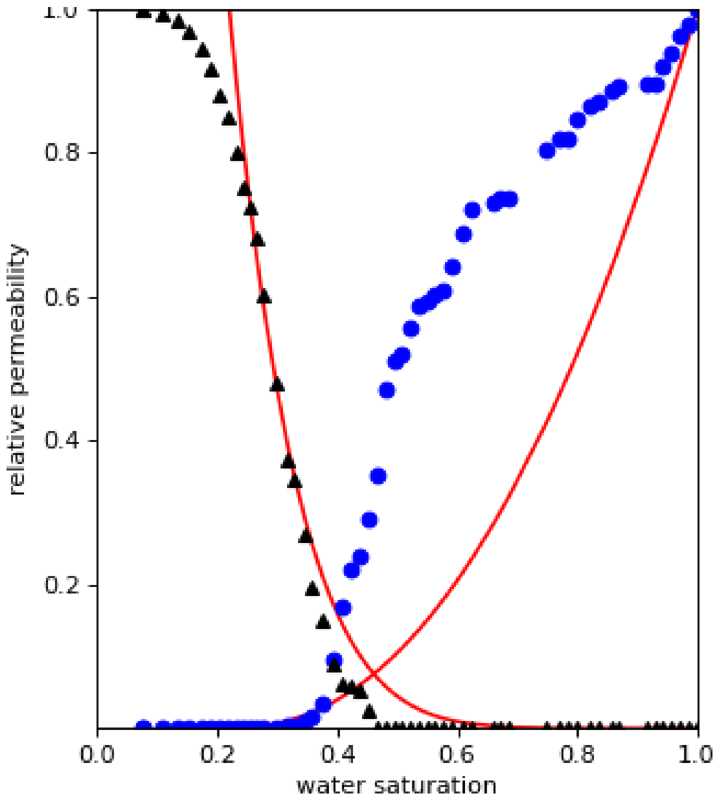


Figure 58: Relative permeability of Berea sandstone and the corresponding Corey match for IFT = 20 mN/m and contact angle of 8 degrees

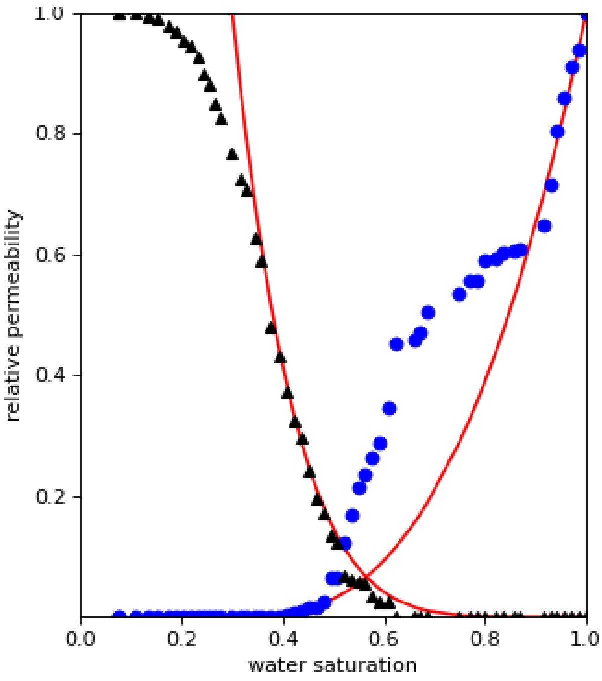


Figure 59: Relative permeability of Berea sandstone and the corresponding Corey match for IFT = 25 mN/m and contact angle of 8 degrees

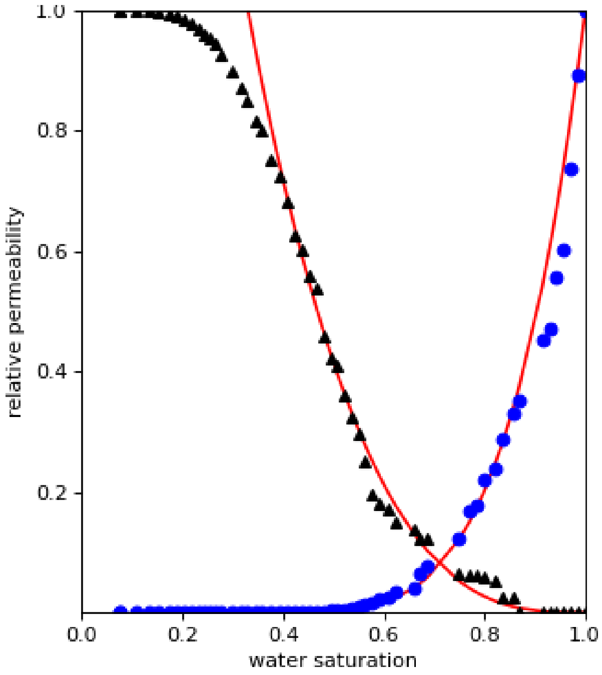


Figure 60: Relative permeability of Berea sandstone and the corresponding Corey match for IFT = 30 mN/m and contact angle of 0 degrees

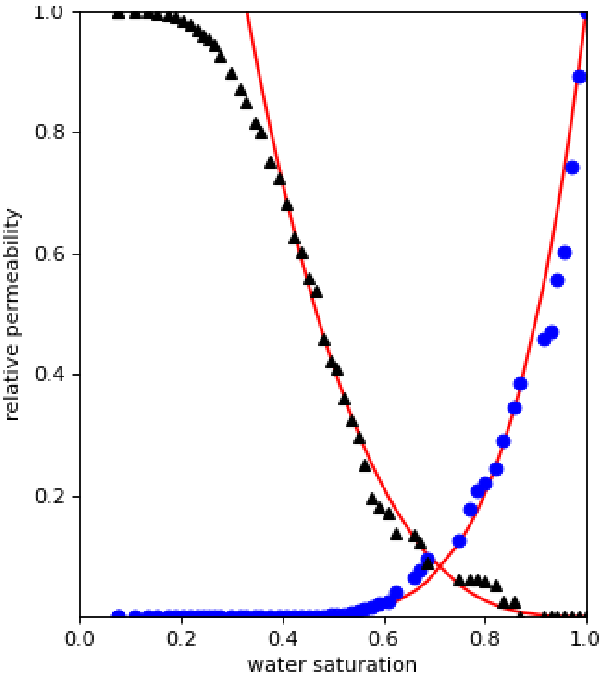


Figure 61: Relative permeability of Berea sandstone and the corresponding Corey match for IFT = 30 mN/m and contact angle of 5 degrees

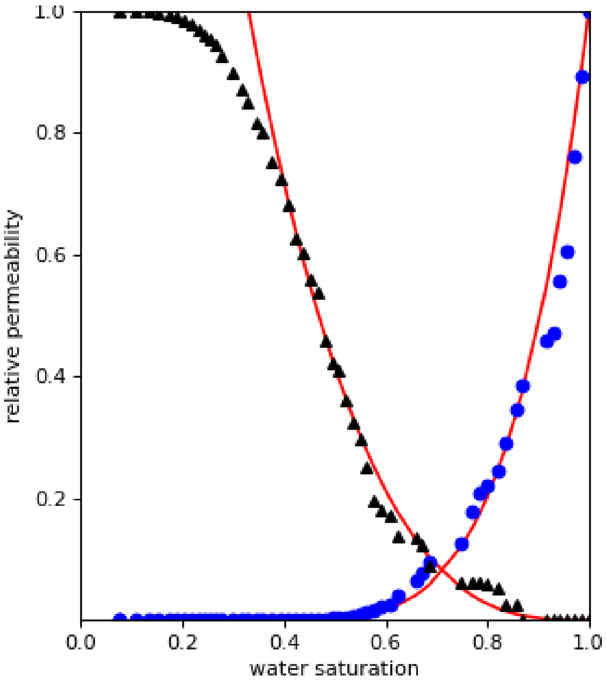


Figure 62: Relative permeability of Berea sandstone and the corresponding Corey match for IFT = 30 mN/m and contact angle of 8 degrees

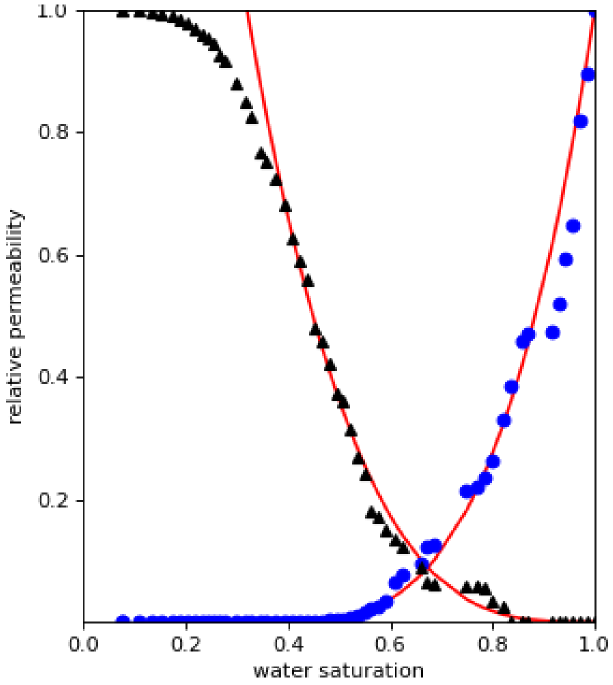


Figure 63: Relative permeability of Berea sandstone and the corresponding Corey match for IFT = 30 mN/m and contact angle of 15 degrees

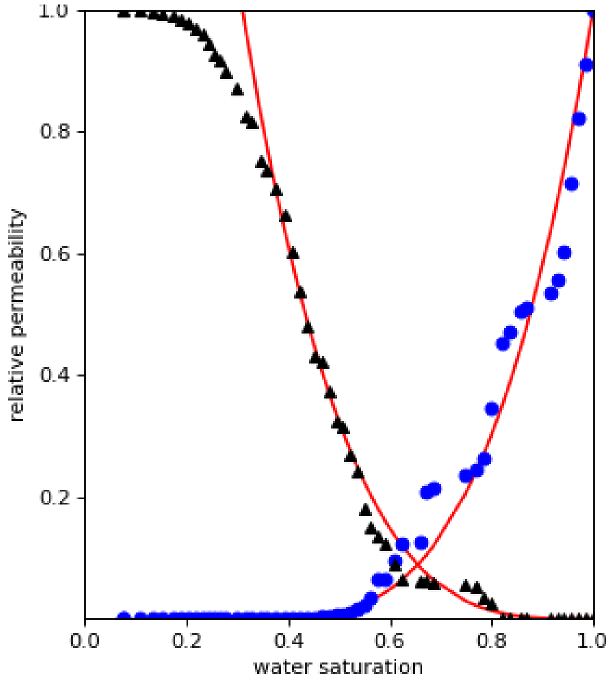


Figure 64: Relative permeability of Berea sandstone and the corresponding Corey match for IFT = 30 mN/m and contact angle of 20 degrees

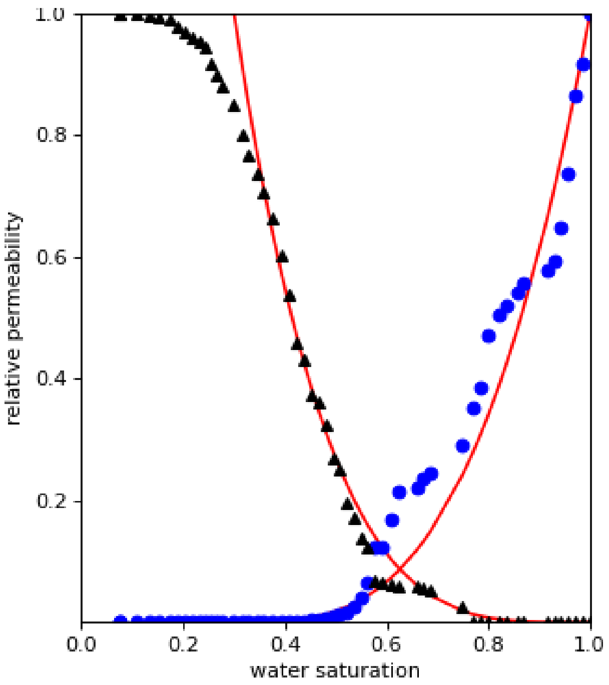


Figure 65: Relative permeability of Berea sandstone and the corresponding Corey match for IFT = 30 mN/m and contact angle of 25 degrees

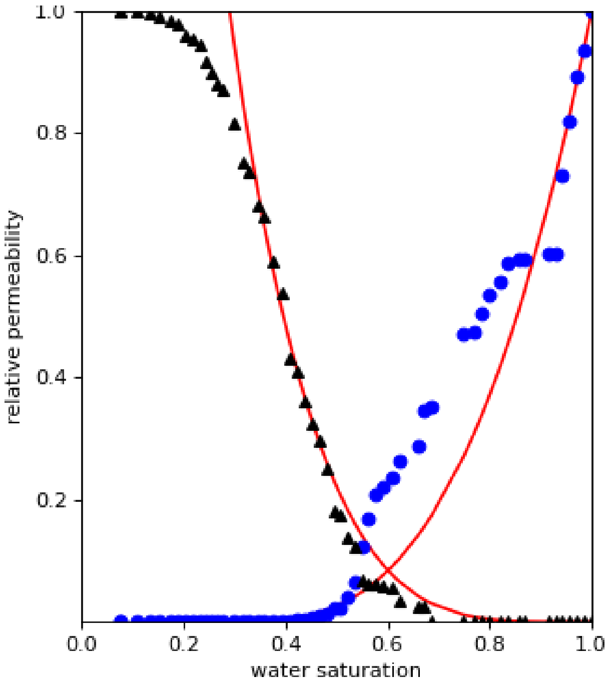


Figure 66: Relative permeability of Berea sandstone and the corresponding Corey match for IFT = 30 mN/m and contact angle of 30 degrees

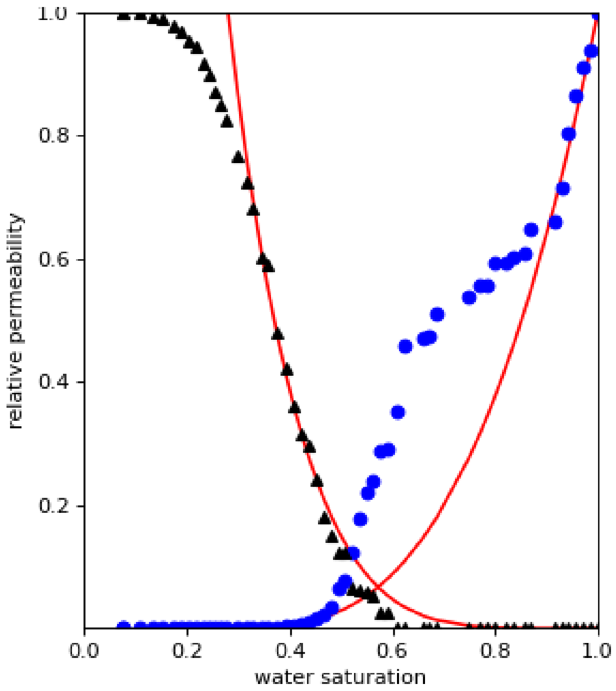


Figure 67: Relative permeability of Berea sandstone and the corresponding Corey match for IFT = 30 mN/m and contact angle of 35 degrees

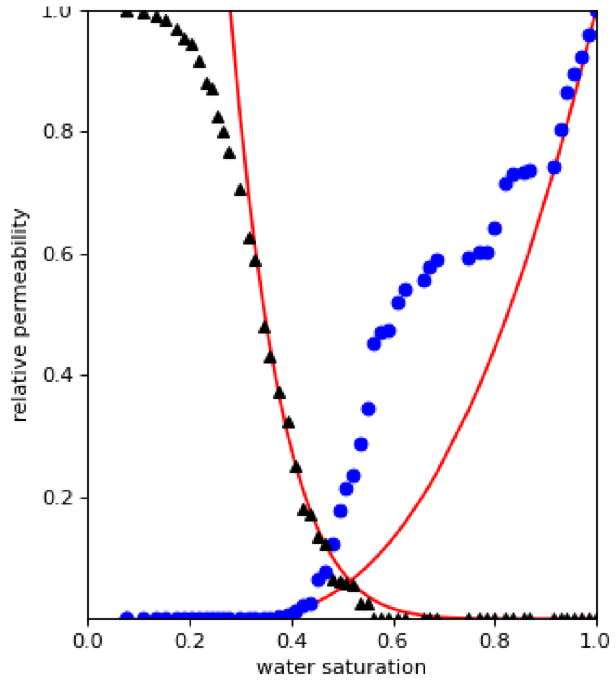


Figure 68: Relative permeability of Berea sandstone and the corresponding Corey match for IFT = 30 mN/m and contact angle of 40 degrees

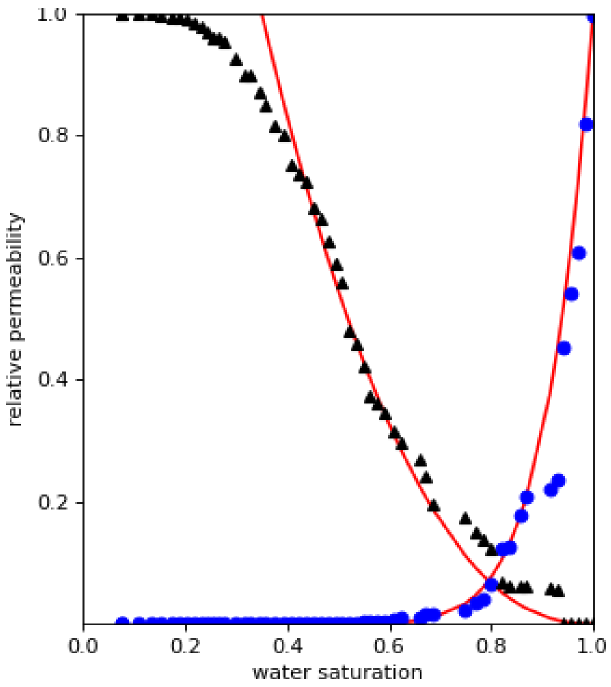


Figure 69: Relative permeability of Berea sandstone and the corresponding Corey match for IFT = 33 mN/m and contact angle of 8 degrees

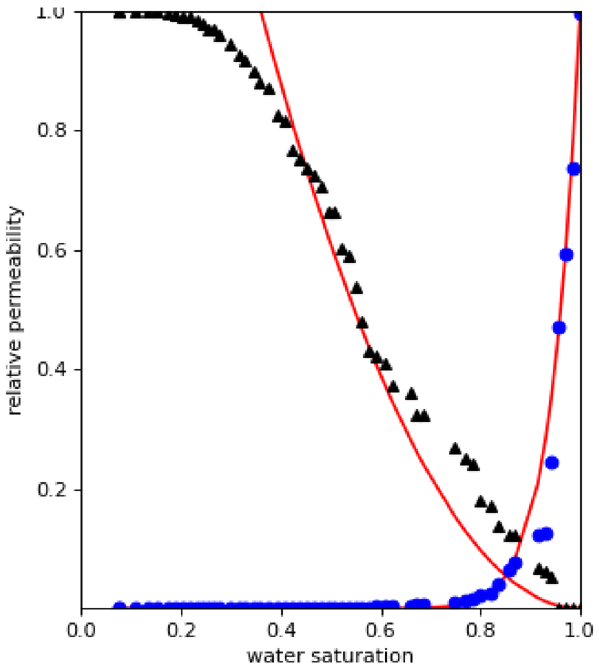


Figure 70: Relative permeability of Berea sandstone and the corresponding Corey match for IFT = 35 mN/m and contact angle of 8 degrees